

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики

Направление подготовки 09.04.01 Информатика и вычислительная техника

Кафедра Информатики и проектирования систем

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка и исследование итеративных параллельных алгоритмов формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных

УДК 004.421.2:517.441:550.83.016

Студент

Группа	ФИО	Подпись	Дата
8BM4Б	Сорокин В.А.		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А.	к.т.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Конотопский В.Ю.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Акулов П.А.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИПС	Демин А.Ю.	к.т.н.		

Томск – 2016 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
Профессиональные компетенции	
P1	Применять глубокие естественнонаучные и математические знания для решения научных и инженерных задач в области информатики и вычислительной техники.
P2	Применять глубокие специальные знания в области информатики и вычислительной техники для решения междисциплинарных инженерных задач.
P3	Ставить и решать инновационные задачи инженерного анализа, связанные с созданием аппаратных и программных средств информационных и автоматизированных систем, с использованием аналитических методов и сложных моделей.
P4	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P5	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P6	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
Универсальные компетенции	
P7	Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
P9	Эффективно работать индивидуально и в качестве члена и руководителя группы, в том числе междисциплинарной и международной, при решении инновационных инженерных задач.
P10	Демонстрировать личную ответственность и ответственность за работу возглавляемого коллектива, приверженность и готовность следовать профессиональной этике и нормам ведения инновационной инженерной деятельности. Демонстрировать глубокие знания правовых, социальных, экологических и культурных аспектов инновационной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению, непрерывному самосовершенствованию в инженерной деятельности, способность к педагогической деятельности.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки 09.04.01 Информатика и вычислительная техника
Кафедра Информатики и проектирования систем

УТВЕРЖДАЮ:
Зав. Кафедрой
_____ Демин А.Ю.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ВМ4Б	Сорокин Василий Александрович

Тема работы:

Разработка и исследование итеративных параллельных алгоритмов формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).

Итеративный параллельный алгоритм формирования функционально-детерминированных структур для классификации и анализа геофизических данных на основе фазовой группировки.

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>Аналитический обзор по литературным источникам методов кластерного анализа и распределенных вычислительных систем; исследование параллельных алгоритмов формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных; разработка распределенной вычислительной системы на основе персональных компьютеров; проведение экспериментов на разных вычислительных мощностях; обсуждение результатов выполненной работы; заключение по работе.</p>
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Общие схемы работы алгоритма; общая функциональная схема распределенной вычислительной системы.</p>

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Доцент Конотопский В.Ю.
Социальная ответственность	Ассистент Акулов П.А.
Раздел на иностранном языке	Сидоренко Т.В.

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Сравнительный обзор и анализ методов структуризации и построения распределенных вычислительных систем

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
--	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Ботыгин И.А.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ4Б	Сорокин Василий Александрович		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки 09.04.01 Информатика и вычислительная техника

Уровень образования Магистратура

Кафедра Информатики и проектирования систем

Период выполнения осенний / весенний семестр 2015/2016 учебного года

Форма представления работы:

Магистерская диссертация

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:

16.06.2016

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.09.2015	Сравнительный обзор и анализ методов структуризации и построения распределенных вычислительных систем	10
10.11.2015	Разработка функциональной структуры основных системных и вычислительных компонентов GRID системы	20
28.01.2016	Разработка основных компонентов распределенной вычислительной системы	40
20.05.2016	Исследование и программные эксперименты по оценке быстродействия в задаче фазовой кластеризации	10
8.06.2016	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
9.06.2016	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры ИПС	Ботыгин И. А.	к.т.н.		10.09.2015

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Информатики и проектирования систем	Демин А.Ю.	к.т.н.		10.09.2015

РЕФЕРАТ

Выпускная квалификационная работа 108 с., 18 рис., 16 табл., 59 источников, 2 прил.

Ключевые слова: фазовая группировка, кластерный анализ, параллельные вычисления, распределенные вычисления.

Объектом исследования являются методы структуризации, а также параллельная реализация алгоритма фазовой группировки.

Цель работы – разработка итеративного параллельного алгоритма формирования функционально-детерминированных структур для классификации и анализа геофизических данных на основе фазовой группировки.

В процессе исследования проводились программные эксперименты по оценке быстродействия в задаче фазовой кластеризации.

В будущем планируется регистрация программного кода для ЭВМ разработанной реализации параллельного алгоритма фазовой группировки метеорологических данных.

Определения, обозначения, сокращения, нормативные ссылки

Грид – форма распределённых вычислений, в которой «виртуальный суперкомпьютер» представлен в виде кластеров, соединённых с помощью сети, слабосвязанных гетерогенных компьютеров, работающих вместе для выполнения огромного количества заданий (операций, работ).

Суперкомпьютер – специализированная вычислительная машина, значительно превосходящая по своим техническим параметрам и скорости вычислений большинство существующих в мире компьютеров.

Вычислительный кластер – группа компьютеров, объединённых высокоскоростными каналами связи, представляющая с точки зрения пользователя единый аппаратный ресурс.

Кластерный анализ – многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы.

Grand challenges – класс фундаментальных научных и инженерных проблем, эффективное решение которых возможно только с использованием мощных вычислительных ресурсов.

Оглавление

Введение.....	11
1. Сравнительный обзор и анализ методов структуризации и построения распределенных вычислительных систем	14
1.1. Обзор методов кластерного анализа.....	14
1.2. Обзор распределенных грид систем	16
2. Исследование и программные эксперименты по оценке быстродействия в задаче фазовой кластеризации	24
2.1. Однопоточное вычисление в стандартном режиме:	24
2.2. Многопоточное вычисление в стандартном режиме	25
2.3. Многопоточное вычисление на базе суперкомпьютера ТПУ	26
2.4. Мультиагентные вычисления на базе распределенной вычислительной системы.	28
3. Функциональная структура основных системных и вычислительных компонентов грид системы	29
3.1. Распределенная вычислительная среда	29
3.2. Алгоритм формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных.	32
4. Технологическая схема реализации и развертывания многомашинной системы для фазовой кластеризации.....	37
4.1. Описание алгоритма кластеризации метеорологических станций по фазе температурного ряда	37
4.1.1. Загрузка входных данных	38
4.1.2. Вычисление фазы температурного ряда.....	39
4.1.3. Алгоритм кластеризации.....	43

4.2. Описание грид системы.....	46
4.2.1. Центр управления	47
4.2.2. Клиентское приложение.....	54
5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	57
5.1. Организация и планирование работы	57
5.1.1 Продолжительность этапов работ	58
5.1.2. Расчет накопления готовности проекта.....	61
5.2. Расчет сметы затрат на выполнение проекта.....	62
5.2.1. Расчет заработной платы	62
5.2.2. Расчет затрат на социальный налог	64
5.2.3. Расчет затрат на электроэнергию	64
5.2.4. Расчет амортизационных расходов.....	65
5.2.5. Расчет прочих расходов	66
5.2.6. Расчет общей себестоимости разработки.....	66
5.2.7. Расчет прибыли	67
5.2.8. Расчет НДС	67
5.2.9. Цена разработки НИР	67
5.3. Оценка экономической эффективности проекта.....	67
5.4. Оценка научно-технического уровня НИР	69
6. Социальная ответственность	71
6.1. Введение	71
6.2. Производственная безопасность	71
6.2.1. Вредные производственные факторы.....	72
6.2.2. Опасные производственные факторы.....	77

6.2.3. Мероприятия и рекомендации по устранению и минимизации	78
6.3. Экологическая безопасность	80
6.4. Безопасность в чрезвычайных ситуациях	82
6.5. Правовые и организационные вопросы обеспечения безопасности	84
6.5.1. Правовые нормы трудового законодательства для рабочей зоны оператора ПЭВМ	84
6.5.2. Организационные мероприятия при компоновке рабочей зоны	85
Заключение	87
Список публикаций студента	89
Список использованных источников	91
Приложение А	99
Приложение Б.....	108

Введение

В последние годы в научных исследованиях и образовании усиливается внимание к использованию высокопроизводительной вычислительной техники. Во многих областях знаний фундаментальные научные исследования связаны с необходимостью проведения масштабных численных экспериментов. Общеизвестным является класс фундаментальных научных и инженерных проблем Grand challenges [1], эффективное решение которых возможно только с использованием мощных вычислительных ресурсов с производительностью сотен Гфлопс ($\sim 10^{12}$ операций в секунду) и выше.

Единственным способом достижения требуемой производительности в настоящее время является использование многопроцессорных вычислительных систем [2] с применением параллельных вычислений.

Список применения параллельных вычислений неуклонно растет, параллельные вычисления применяются в областях, связанных с проведением больших расчетов. Так, в системах поддержки проектирования (CAD – Computer aided design) необходимость осуществлять моделирование в реальном масштабе времени предъявляет высокие требования к производительности программного обеспечения. В инженерных приложениях и математическом моделировании для повышения точности используются параллельные алгоритмы. Также, большой областью применения являются бизнес-приложения, например, задачи, связанные с анализом финансовых рынков и прогнозирования курсов валют. В итоге можно с уверенностью утверждать, что параллельные информационные технологии превратились из узконаправленной дисциплины в необходимую составляющую комплекса знаний разработчика современного программного обеспечения [3].

Идея распараллеливания вычислений основана на том, что большинство задач может быть разделено на набор меньших задач, которые могут быть решены одновременно. Основная цель параллельных вычислений – уменьшение времени решения задачи. Для решения трудоемких задач такие

вычисления могут быть запущены, например, на суперкомпьютерах, вычислительных кластерах или грид-системах.

Суперкомпьютер – специализированная вычислительная машина, значительно превосходящая по своим техническим параметрам и скорости вычислений большинство существующих в мире компьютеров. Как правило, современные суперкомпьютеры представляют собой большое число высокопроизводительных серверных компьютеров, соединенных друг с другом локальной высокоскоростной магистралью [4]. Стоимость таких систем очень высока.

Вычислительные кластеры – группа компьютеров, объединенных высокоскоростными каналами связи, представляющая с точки зрения пользователя единый аппаратный ресурс [5].

Грид-вычисления – форма распределенных вычислений, в которой «виртуальный суперкомпьютер» представлен в виде кластеров, соединенных с помощью сети, слабосвязанных гетерогенных компьютеров, работающих вместе для выполнения огромного количества задания. Грид-вычисления можно организовать на базе множества устаревших моделей персональных компьютеров объединенных в иерархическую локальную вычислительную сеть [6]. Распределенные грид-системы являются разновидностью параллельных вычислений, которые основываются на обычных компьютерах и выгодно отличается от обычных суперкомпьютеров тем, что могут быть развернуты при помощи обычных протоколов, например, Ethernet. Таким образом обычные компьютеры могут быть объединены в единую распределенную систему, а их ресурсы могут быть использованы для вычислений.

Одной из областей, где возникают фундаментальные научные или инженерные задачи с широкой областью применения, эффективное решение которых возможно только с использованием мощных (суперкомпьютерных) вычислительных ресурсов является предсказание погоды, климата и глобальных изменений в атмосфере [1]. Примером такой задачи, в частности, является формирование скрытых функционально-детерминированных структур

для классификации и анализа метеорологических данных. В данном случае, необходимость реализации возможности параллельных вычислений обусловлена огромным количеством данных. Число элементов дискретизации метеорологических данных только для одной точки Земли за 50 лет может достигать несколько тысяч, а количество таких точек для анализа климатических изменений превышает несколько сотен. Время, необходимое для анализа такого большого объема данных может доходить до нескольких дней, что подчеркивает необходимость и важность параллельных вычислений для данной задачи.

Данная работа посвящена разработке и исследованию итеративного параллельного алгоритма, а также построению грид-системы для выполнения данного алгоритма.

1. Сравнительный обзор и анализ методов структуризации и построения распределенных вычислительных систем

1.1. Обзор методов кластерного анализа

Одним из подходов к обработки больших объем геофизических данных (в частности, метеорологических данных) является кластерный анализ.

Кластеризация (или кластерный анализ) – это задача разбиения множества объектов на группы, называемые кластерами.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости – нормализация значений переменных.
3. Вычисление значений меры сходства между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

По способу обработки данных методы кластерного анализа можно разделить на иерархические, неиерархические. Иерархические методы выполняют последовательное объединение меньших кластеров в большие (агломеративные) или разделяют большие кластеры на меньшие (дивизивные) [7].

Также можно выделить некоторые особенности методов, например, по способу анализа данных: четкие, нечеткие, а также по количеству применений алгоритмов кластеризации: с одноэтапной кластеризации, с многоэтапной кластеризацией [8].

Алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) предусматривает двухэтапный процесс кластеризации и используется для очень больших наборов данных. К ограничениям этого алгоритма относится то, что он работает только с числовыми данными.

K-means (метод k-средних) разбивает множество элементов векторного пространства на заранее известное число кластеров k . Алгоритм прост в использовании, однако чувствителен к выбросам, которые могут исказить среднее, а также к его недостаткам можно отнести необходимость знания количества кластеров [9].

Алгоритм CLOPE (Clustering with slope) предназначен для работы с транзакционными данными, обладает высокой масштабируемостью и скоростью работы, так же качеством кластеризации. CLOPE автоматически подбирает необходимое количество кластеров и широко применяется в Data mining [10].

Алгоритм PAM (Partitioning around medoids) является модификацией алгоритма k-средних. Алгоритм менее чувствителен к шумам и выбросам данных, чем алгоритм k-средних. PAM эффективен для небольших баз данных, его не следует использовать для больших наборов данных [11]. Применяется в широком спектре предметных областей, таких как анализ текстов, биоинформатика, интеллектуальные транспортные системы [12].

Самоорганизующиеся карты Кохонена – одна из разновидностей нейросетевых алгоритмов, использующие метод без учителя. Представляет собой один из вариантов кластеризации многомерных векторов. Этот алгоритм можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью. Данную методику можно использовать для поиска и анализа закономерностей в исходных данных [13].

Таким образом, не существует единого универсального алгоритма кластеризации. При использовании любого алгоритма важно понимать его достоинства и недостатки, учитывать природу данных, с которыми он лучше работает и способность к масштабируемости.

Проблема кластеризации Больших Данных состоит в том, что имеющиеся алгоритмы предполагают возможность непосредственного обращения к любой информационной сущности в исходных данных (заранее невозможно предугадать, какие именно сущности понадобятся алгоритму). В свою очередь, исходные данные могут быть распределены по разным серверам, и при этом не гарантируется, что каждый кластер хранится строго на одном сервере. Если распределение данных по серверам делать прозрачным для алгоритма кластеризации (он считает, что данные расположены в некоторой распределенной виртуальной памяти), то это неизбежно приведет к копированию больших объемов с одного сервера на другой.

Решение проблемы может быть следующим. На каждом сервере запускается свой алгоритм, который оперирует только данными этого сервера, а на выходе дает параметры найденных кластеров и их веса, оцениваемые исходя из количества элементов внутри кластера. Затем полученная информация собирается на центральном сервере и производится метакластеризация – выделение групп близко расположенных кластеров с учетом их весов. Этот метод универсален, хорошо распараллеливается и может использовать любые другие алгоритмы кластеризации, однако он требует проведения серьезных научных исследований, тестирования на реальных данных и сравнения полученных результатов с другими «локальными» методами.

Таким образом, для анализа Больших Данных подавляющая часть методов кластеризации неприменима в чистом виде и необходимы дополнительные исследования и большие вычислительные мощности [14].

1.2. Обзор распределенных грид систем

Одним из перспективных и недорогих способов организации метакомпьютинга является объединение автономных вычислительных ресурсов (компьютеров) в единый виртуальный вычислительный узел (грид). Распределенные грид-системы являются разновидностью параллельных вычислений, которые основывается на обычных компьютерах и выгодно

отличается от обычных суперкомпьютеров тем, что могут быть развернуты при помощи обычных протоколов, например, Ethernet. Таким образом обычные компьютеры могут быть объединены в единую распределенную систему, а их ресурсы могут быть использованы для вычислений.

Одной из важных характеристик распределенных систем состоит в том, что от пользователей скрыты различия между компьютерами и способы связи между ними. Другой важной характеристикой является способ, при помощи которого пользователи и приложения единообразно работают в распределенных системах, независимо от того, где и когда происходит их взаимодействие. Распределенные системы должны также относительно легко поддаваться расширению, или масштабированию [15].

К основным отличительным признакам распределенных систем относят:

- Отсутствие единого времени для компонентов распределенной системы характеризует территориальное распределение компонентов системы, а именно процессоров, входящих в ее состав, из чего следует отсутствие синхронности в их работе;
- Отсутствие общей памяти. Ключевая характеристика, из которой следует необходимость обмена сообщениями между программными компонентами распределенной системы для их взаимодействия и синхронизации.
- Географическое распределение;
- Независимость и гетерогенность. Компьютеры, входящие в состав распределенной системы слабо связаны в том смысле, что они могут иметь различный состав и различную производительность и, следовательно, обеспечивать различное время выполнения идентичных задач. Обычно они не являются частями одной специализированной системы, но функционируют совместно, предоставляя свои службы друг другу для выполнения общей задачи [16].

Основная задача распределенных систем – облегчить пользователям доступ к удаленным ресурсам и обеспечить их совместное использование,

регулируя этот процесс [15]. Эффективная распределенная система должна обладать следующими свойствами: прозрачность (англ. *transparency*), открытость (англ. *openness*), безопасность (англ. *security*), масштабирование (англ. *scalability*) [16].

Под **прозрачностью** распределенной системы понимают ее способность скрывать свою распределенную природу, а именно, распределение процессов и ресурсов по множеству компьютеров, и представляться для пользователей и разработчиков приложений в виде единой централизованной компьютерной системы. Для того чтобы распределенная вычислительная система могла быть представлена пользователю как единая система, применяют следующие типы прозрачности:

- Прозрачный доступ к ресурсам – от пользователей должна быть скрыта разница в представлении данных и в способах доступа к ресурсам вычислительной системы;
- Прозрачное местоположение ресурсов – место физического расположения требуемого ресурса должно быть несущественно для пользователя;
- Репликация – сокрытие от пользователя того, что в реальности существует более одной копии используемых ресурсов;
- Параллельный доступ – возможность совместного (одновременного) использования одного и того же ресурса различными пользователями независимо друг от друга. При этом факт совместного использования ресурса должен оставаться скрытым от пользователя;
- Прозрачность отказов – отказ (отключение) каких-либо ресурсов системы не должны оказывать влияния на работу пользователя и его приложения [17].

Открытая распределенная система – это система, предлагающая службы, вызов которых требует стандартные синтаксис и семантику. В распределенных системах службы обычно определяются через интерфейсы. Будучи правильно описанным, определение интерфейса допускает возможность

совместной работы произвольного процесса, нуждающегося в таком интерфейсе, с другим произвольным процессом, предоставляющим этот интерфейс. Определение интерфейса также позволяет двум независимым группам создать абсолютно разные реализации этого интерфейса для двух различных распределенных систем, которые будут работать абсолютно одинаково. Также, открытая система должна обладать: *способностью к взаимодействию*, что характеризует, насколько две реализации систем или компонентов от разных производителей в состоянии совместно работать, полагаясь только на то, что службы каждой из них соответствуют общему стандарту; *переносимостью*, что характеризует то насколько приложение, разработанное для распределенной системы А, может без изменений выполняться в распределенной системе В, реализуя те же, что и в А интерфейсы; *гибкостью* – легкость конфигурирования системы, состоящей из различных компонентов [15].

Масштабируемость – способность вычислительной системы эффективно справляться с увеличением числа пользователей или поддерживаемых ресурсов без потери производительности и без увеличения административной нагрузки на ее управление. Систему называют масштабируемой, если она способна увеличивать свою производительность при добавлении новых аппаратных средств.

Масштабируемость распределенных систем определяют по следующим направлениям:

- Нагрузочная масштабируемость – способность системы увеличивать свою производительность при увеличении нагрузки путем замены существующих аппаратных компонентов (вертикальное масштабирование) на более мощные или путем добавления новых аппаратных средств (горизонтальное масштабирование);
- Географическая масштабируемость – способность системы сохранять свои основные характеристики, такие как производительность, простота и удобство использования, при

территориальном разнесении ее компонентов от более локального взаимного расположения до более распределенного;

- **Административная масштабируемость.** Характеризует простоту управления системой при увеличении количества административно независимых организаций, обслуживающих части одной распределённой системы.

Концепция грида появилась не как абстрактная идея, а как ответ на появляющиеся потребности в крупных информационно-вычислительных ресурсах, динамически выделяемых для решения громоздких задач, в научной, индустриальной, административной и коммерческой областях деятельности. Создание грид-среды подразумевает распределение вычислительных ресурсов по территориально разделенным узлам, на которых установлено специализированное программное обеспечение для того, чтобы распределять задания по узлам и принимать их там, возвращать результаты пользователю, контролировать права пользователей на доступ к тем или иным ресурсам, осуществлять мониторинг ресурсов, и так далее. Общедоступные ресурсы могут включать:

- **Вычислительные ресурсы** предоставляют задаче пользователя грид-системы процессорные мощности. Вычислительными ресурсами могут быть как кластеры, так и отдельные рабочие станции. Основной характеристикой вычислительного ресурса является производительность;
- **Ресурсы хранения** используют прикладное программное обеспечение, реализующее унифицированный интерфейс управления и передачи данных. Основной характеристикой ресурсов хранения данных является их объем;
- **Информационные ресурсы** и каталоги являются особым видом ресурсов хранения данных. Они служат для хранения и предоставления метаданных и информации о других ресурсах грид-системы. Информационные ресурсы позволяют структурированно

хранить огромный объем информации о текущем состоянии грид-системы и эффективно выполнять задачи поиска ресурсов;

- **Сетевой ресурс** является связующим звеном между распределенными ресурсами грид-системы. Основной характеристикой сетевого ресурса является скорость передачи данных [18].

К основным областям применения грид-систем относят научные исследования, аэрокосмическая и автомобильная промышленность, архитектура, электроника, энергетика, финансово-банковская инфраструктура, медицина и биотехнологии, производство, медиа-развлечения [19].

GRID – популярная в настоящее время технология, на базе подходов GRID было создано много приложений, большинство из них имеют ярко выраженную математическую направленность (например, решение больших вычислительных задач методом разделения их на малые части). Частным случаем таких систем является Desktop-Grid – грид-системы, в которых в роли вычислительных узлов выступают обычные персональные компьютеры.

Одним из примером является свободно распространяемая система Condor, ориентированная на работу с не отчуждаемыми (разделяемыми с владельцем компьютера) ресурсами. Система Condor обладает механизмами организации очереди работ, политики планирования, назначения приоритетов, учета использования и управления ресурсами, поддержки контрольных точек, рестарта и миграции заданий. В Condor развит весьма мощный язык описания ресурсов, позволяющий формально описать как требования к типам и объемам ресурсов со стороны задания, так и ограничения на доступ к ресурсам со стороны владельцев [20].

Также, существует система метакомпьютинга X-Com предназначена для быстрого развертывания и проведения распределенных расчетов. Система представляет собой инструментарий для адаптации и поддержки выполнения программ в распределенных неоднородных средах. В основе архитектуры X-Com лежит клиент-серверная схема. Соответственно ей, прикладная задача

должна быть логически разделена на две части: серверную и клиентскую. Серверная часть отвечает за разбиение задачи на множество независимых порций и объединение результатов [21].

В общем виде при организации подобных инфраструктур можно выделить четыре подхода: первый основан на создании проектов, к которым подключаются исполнительные компьютеры; второй подход состоит в применении P2P-технологий (Peer-to-Peer) и объединении исполнительных компьютеров в одноранговые сети; третий подход характеризуется системами с централизованным управлением; четвёртый подход представлен частными разработками корпоративных систем.

Так, платформа BOINC, предназначенная для поддержки проектов, не предусмотрена для запуска приложений, т.к. для каждого нового приложения необходимо создавать свою программно-аппаратную инфраструктуру [22].

OurGrid – открытая, свободная к объединению, совместная сеть, которой лаборатории жертвуют свои неактивные вычислительные ресурсы в обмен на доступ к неактивным ресурсам других лабораторий, когда необходимо. Используется одноранговая технология, которая позволяет каждой лаборатории взаимодействовать с системой [23].

XtremWeb представляет собой программное обеспечение с открытым кодом для создания грид-систем из персональных компьютеров. В настоящее время дальнейшее развитие системы ведется под именем XWNER. Архитектура XWNER состоит из трех основных типов компонентов: серверы, рабочие и клиенты. Серверы обеспечивают функционирование центральных сервисов системы, рабочие устанавливаются владельцами ресурсов на своих компьютерах для предоставления данных ресурсов в рамках системы. Клиенты устанавливаются пользователями ресурсов на своих компьютерах для взаимодействия с системой и использования агрегированных ею ресурсов. В системе реализованы механизмы прав доступа как на уровне пользователя, так и групп пользователей [24].

Слабыми местами систем, реализующих подход с централизованным управлением (например, X-Com), является планирование, которое не позволяет обеспечить выполнение задания в заранее указанный пользователем срок. Необходимость предварительной подготовки задания в X-Com также сужает область применения этой системы. Кроме того, в этих разработках, а также в системах CCOF и OurGrid, представляющих P2P-подход, не используются ставшие фактическими стандартами протоколы грида. Это приводит к тому, что такого рода разработки не являются интероперабельными с другими грид-системами. То же самое можно сказать практически обо всех корпоративных решениях.

2. Исследование и программные эксперименты по оценке быстродействия в задаче фазовой кластеризации

Как было сказано выше, необходимость параллельных вычислений задачи анализа температурных данных обуславливается огромным количеством данных. В данной работе были проведены исследования на двух наборах данных: среднемесячная температура за 56 лет на 818 станциях (количество измерений на одну станцию составляет 660, общее число измерений больше пятисот тысяч, объем данных 2,5 мегабайта) и среднесуточная температура за 55 лет на 818 станциях (количество измерений на одну станцию составляет более 19 тысяч, общее число измерений более 16 миллионов, объем данных 77 мегабайт). Для проведения высокоточных мелкомасштабных вычислений в климатических задачах требуются ряды метеорологических параметров с интервалом регистрации секунды, а в некоторых случаях и миллисекунды, что увеличивает объемы обрабатываемых данных на несколько порядков.

Для анализа эффективности вычислений был проведен ряд программных экспериментов на различных вычислительных мощностях:

- однопоточное вычисление в стандартном режиме (desktop);
- многопоточное вычисление в стандартном режиме (desktop);
- многопоточное вычисление на базе суперкомпьютера ТПУ;
- мультиагентные вычисления на базе распределенной вычислительной системы.

2.1. *Однопоточное вычисление в стандартном режиме:*

Для проведения экспериментов использовались следующие вычислительные мощности:

Персональный компьютер: Процессор Intel Core i5 3317U (4x2400 Mhz), 8 гигабайт оперативной памяти. За основу вычислительного процесса была взята схема итеративного параллельного алгоритма формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных. Подробное описание алгоритмов вычислений и их

реализация изложено в главах 3 и 4. Особенностью практической реализации алгоритма в настоящем эксперименте являлось отсутствие одновременно выполняющихся (параллельных) частей.

Для первого набора данных были получены следующие результаты вычислений: 27 секунд.

Для второго набора данных были получены следующие результаты вычислений: 13 минут 41 секунда.

2.2. Многопоточное вычисление в стандартном режиме

Для проведения экспериментов использовались следующие вычислительные мощности:

Персональный компьютер: Процессор Intel Core i5 3317U (2x2400 Mhz), 8 гигабайт оперативной памяти. В данной конфигурации персонального компьютера используется технология Hyper-threading – технология, разработанная компанией Intel, которая реализует идею «одновременной мультипоточности». В этом случае один физический процессор (одно физическое ядро) определяется операционной системой как два отдельных процессора, то есть, два логических ядра. Таким образом, вычислительная мощность в эксперименте для параллельных вычислений обеспечивалась четырьмя потоками.

Для экспериментов с многопоточным вычислением алгоритма были использованы разное количество потоков: 4,40,400,818 потоков. Полученные результаты приведены в таблице 1.

Таблица 1. Полученные результаты многопоточного вычисления

		4 потока	40 потоков	400 потоков	818 потоков
ПК	Набор данных 1	18 секунд	16 секунд	15 секунд	15 секунд
	Набор данных 2	10 минут 46 секунд	10 минут 8 секунд	11 минут 9 секунд	11 минут 16 секунд

По сравнению с однопоточным вычислением (эксперимент 1) прирост производительности вычислений при многопоточном варианте составил 25-30%.

Типовая картина загруженности ядер при выделении потоков больше, чем 40 приведена на рисунке 1. Эксперимент был проведен на машине, оснащенной процессором Intel Core i7-2600 (4*3.40 Ghz), использующий технологию hyper-threading с четырьмя логическими и четырьмя физическими ядрами.

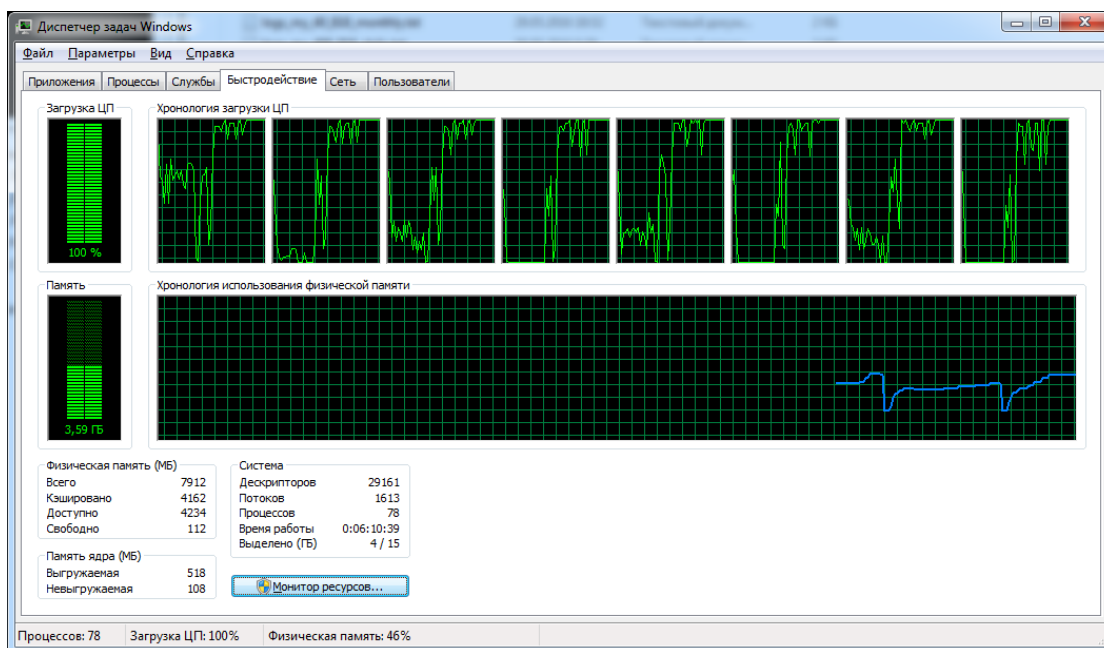


Рисунок 1. Типовая картина загруженности ядер

2.3. Многопоточное вычисление на базе суперкомпьютера ТПУ

К сожалению, в данном эксперименте из декларируемых вычислительных возможностей вычислительного узла Blade-сервера SBI-7126T-T1E (комплекс вычислительный кластерный «СКИФ №2»), характеристики которого приведены ниже, в полном объеме предоставлены не были.

Характеристики «СКИФ №2»:

- Количество вычислительных узлов: 39;
- Количество процессоров: 78 (Nehalem 4C X5570 2.93G 8M 6.4G);

- Количество вычислительных ядер: 320 (2.9Ghz);
- Общий объем ОП: 479ГБ;
- Системная сеть: Infiniband 4x, 39 портов;
- Вспомогательная сеть: Gigabit Etherhet, 39 портов;
- Пиковая производительность: 3.01ТФЛОПС.

Приведенный протокол работы (таблица 2) показал, что эффективность реально выделенных мощностей оказалась соизмеримой с вычислительными мощностями персонального компьютера в эксперименте 1.

Таблица 2. Протокол работы на «СКИФ-2»

```
Total phase calculation time: 21
Total correlation table calculation time: 15
Total groups allocation time: 0
Total typical calculation time: 6
Total correlation table calculation time: 16
Total groups allocation time: 0
Total typical calculation time: 7
Total correlation table calculation time: 15
Total groups allocation time: 0
Total typical calculation time: 7
Total correlation table calculation time: 16
Total groups allocation time: 0
Total typical calculation time: 7
Total correlation table calculation time: 17
Total groups allocation time: 0
Total typical calculation time: 6
Total correlation table calculation time: 17
Total groups allocation time: 0
Total typical calculation time: 7
Total correlation table calculation time: 19
Total groups allocation time: 0
Total typical calculation time: 7
Total correlation table calculation time: 16
Total groups allocation time: 0
Total typical calculation time: 7
Total correlation table calculation time: 16
Total groups allocation time: 0
Total typical calculation time: 7
Total correlation table calculation time: 16
Total groups allocation time: 0
Total typical calculation time: 4
```

Total correlation table calculation time: 18
Total groups allocation time: 0
Total typical calculation time: 4
Total time: 283
Total loops: 11

2.4. Мультиагентные вычисления на базе распределенной вычислительной системы.

Для вычислений с помощью разработанной грид системы использовалось 6 персональных компьютеров с процессорами Intel Core 2 Duo (2x3 Ghz) с 2 гигабайтами оперативной памяти. Один из компьютеров выступал в качестве командного центра, остальные пять предоставляли свои вычислительные мощности для расчетов.

Время расчета одной итерации для первого набора данных составило 17 минут 31 секунд, при этом время выполнения операций с файлами (открытие, считывание, запись, передача) составило 16 минут 19 секунд. Время выполнения расчетов составляет 72 секунды, в которое также входит время операций с файлами на узлах.

Такой результат показывает, что для данного типа задач вычисления с помощью грид системы, построенной на обычных персональных компьютерах, не подходит, так как время подготовки вычислений значительно превышает время самих вычислений. Более подходящей является задача, для которой необходимо большое количество вычислений над одними и теми же данными.

3. Функциональная структура основных системных и вычислительных компонентов грид системы

3.1. *Распределенная вычислительная среда*

Как было отмечено выше, существует большое множество распределенных вычислительных систем, главным различием которых является контроль и управление всей системой. Так, существуют разные виды централизованных распределенных систем (кластеры, клиент-серверные системы) и децентрализованных систем (грид-системы, облачные вычисления). Распределенным децентрализованным систем присуща более высокая степень модульности. Они содержат множество компонентов, которые также могут быть разбиты на части. Эти компоненты являются автономными, в любой момент времени они имеют полный контроль над своими частями.

Разработанная в настоящей выпускной квалификационной работе вычислительная среда представляет собой горизонтально масштабируемый кластер распределенных в пространстве серверов исследователей со своими вычислительными узлами-агентами обработки.

На первом этапе, на управляющем компьютере устанавливается центр управления. Именно на центр управления возложена организация всех вычислительных процессов в среде. Центр управления вычислениями координирует функциональную обработку данных, руководствуясь инструкциями (по данным и вычислительному функционалу), полученными от исследователей.

Узлом вычисления в разрабатываемой системе называется любой пользовательский компьютер с установленным клиентским приложением, готовый предоставить свои вычислительные ресурсы для обработки данных. Центр управления в каждый момент времени осуществляет мониторинг доступности и занятости каждого узла. Если в очереди задач есть задача, нуждающаяся в обработке, центр назначает свободному узлу ее выполнение.

Для анализа данных сами данные должны храниться и выдаваться отдельным узлам по запросу, причем узлу не нужно иметь представление обо всем объеме данных, ему необходима только та информация, которую необходимо обработать. Все данные в системе хранятся в хранилище метеоданных, в том числе и промежуточные результаты вычислений, полученные от клиентских приложений, запущенных на вычислительных узлах. За распределение, сбор и конечный анализ данных отвечает модуль центра управления, который разбивает общие данные на порции, необходимые узлу в нужный момент времени.

Назначение задачи для выполнения происходит путем направления на свободный узел исполняемого программного кода и необходимого блока данных для вычислений. После успешного выполнения задачи узел отправляет результаты центру управления.

За формирование очереди задач и контроль ее соблюдения отвечает специальный компонент программного обеспечения – диспетчер задач. Так, формируется список подлежащих выполнению задач, которые могут быть выполнены в данный момент, а также очередь ожидания, куда помещаются задачи, которые не могут быть выполнены до определенных этапов.

Схема взаимодействия компонентов системы изображена на рисунке 2.

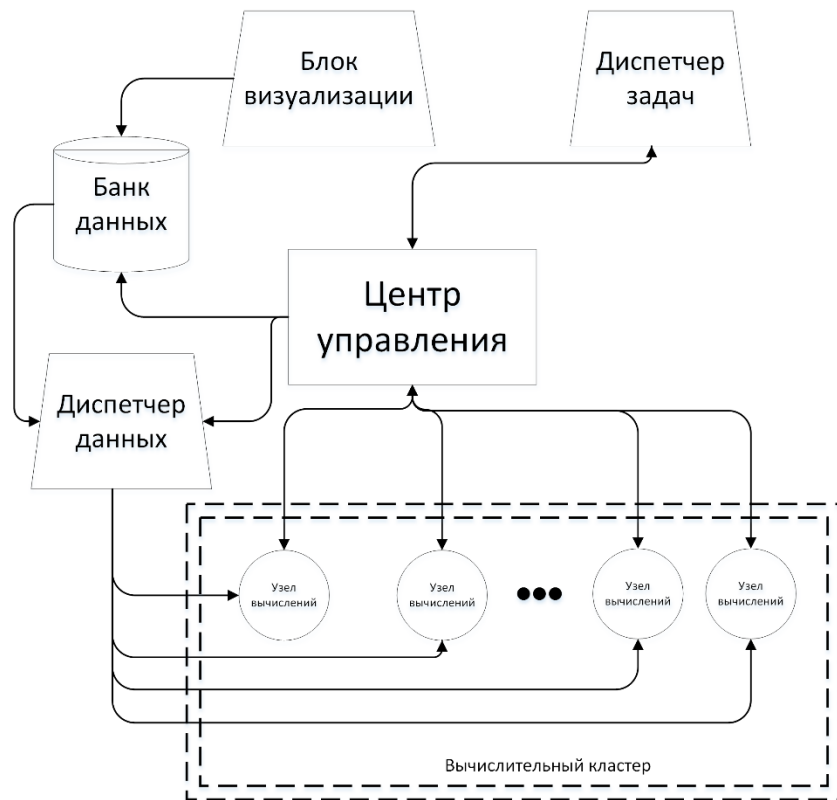


Рисунок 2. Схема взаимодействия компонентов

В процессе подготовки задачи к вычислению с помощью грид-системы пользователь формирует карту задач в виде графа, на которой указываются задачи и их зависимости между собой. Так, пользователь указывает связи тех задач, которым необходимо дождаться выполнения предыдущего этапа вычислений (рис. 3Б).

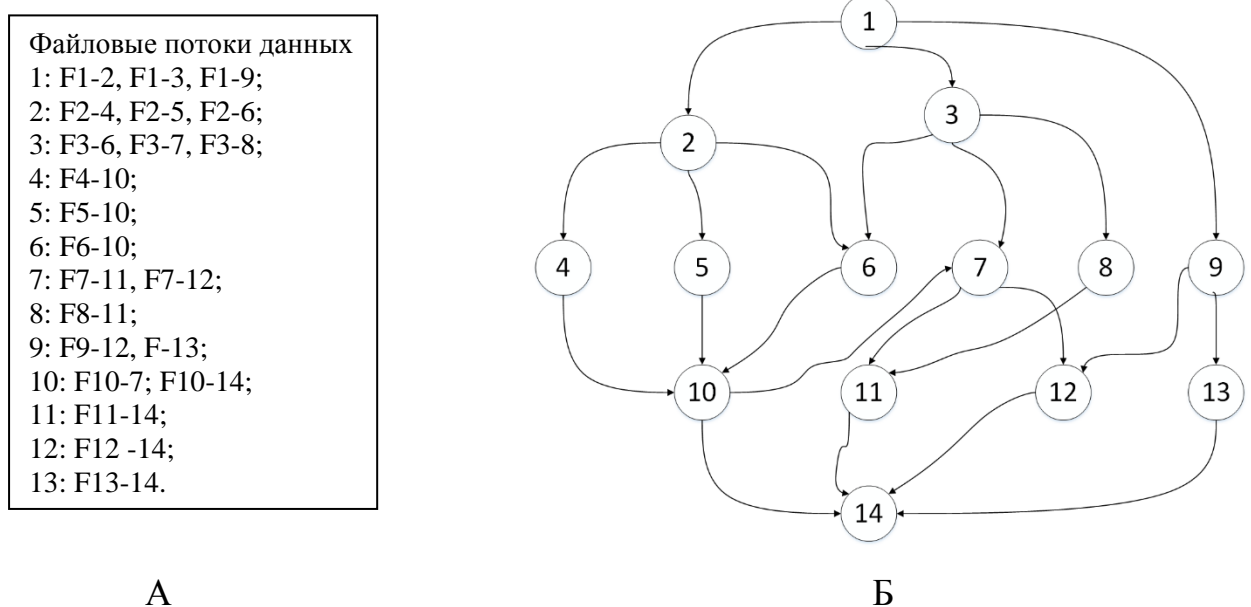


Рисунок 3. Пример графа вычислений

В процессе формирования карты-графа пользователю необходимо сформировать для каждого узла функциональный модуль вычислений (исполняемый программный код), который должен быть выполнен, а также набор данных, с которыми будут производиться вычисления. Заметим, что каждая вершина карты-графа формирует именные файловые потоки данных, необходимые для работы последующих модулей вычислений (рис. 3А).

3.2. Алгоритм формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных.

В настоящее время, в связи с глобальными изменения климата, большой интерес вызывает изучение многомерных геофизических данных и выявление в них закономерностей. Изменения природно-климатических процессов, как правило, имеют циклический характер и, следовательно, в полной мере могут исследоваться методами спектрального анализа сигналов. Базовой характеристикой колебательного процесса является фаза. В отличие от амплитуды, характеризующей величину изменяемого параметра, фаза отражает информацию о периодичности этого изменения во времени. На основе фазы можно строить новые оценки, определяющие законы поведения исследуемых параметров за определенные временные интервалы, а также применять методы и алгоритмы кластерного анализа [25,26].

Исследование среднемесячных температурных рядов, полученных за 55 лет на 333 метеостанциях, показало, что изменения таких данных за исследуемый период формируют колебательный процесс с квазипериодом в один год. Для описания сезонных изменений представляют интерес отклонения температуры от годового цикла. Однако их непротиворечивое выделение затруднено [27]. Исследования показали, что частотный спектр температурного колебательного процесса является узкополосным и для большинства метеостанций имеет одну моду, что позволяет использовать условия причинности и ввести фазу колебания [28]. Годовая составляющая фазы представляет собой линейную функцию, которая удалялась из фазы на

интервале 55 лет. Оставшиеся флуктуации фазы, после удаления из фазы годовой составляющей, позволяют непротиворечиво исследовать их согласованное поведение или синхронность, применяя в виде меры этой характеристики коэффициент корреляции по Пирсону.

Для расчета фаз каждого температурного ряда был реализован алгоритм, обобщенная схема которого представлена на рисунке 4.

Этап 1: Быстрое преобразование Фурье. Для представления цифрового ряда в виде суммы выбранных косинусоидальных и синусоидальных функций (переход из временного пространства в частотное) используется дискретное преобразование Фурье (в нашем случае БПФ).

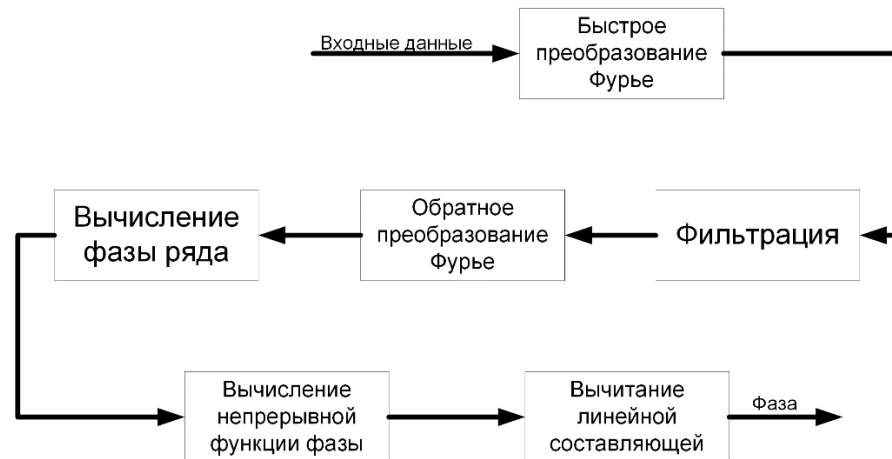


Рис. 4. Общая схема алгоритма расчета фазы температурного ряда

Этап 2: Фильтрация. Для подавления шума используется медианная фильтрация. Одномерный медианный фильтр основан на эвристическом методе обработки и представляет собой скользящее окно, охватывающее нечетное число элементов.

Этап 3: Обратное преобразования Фурье. Для возвращения из частотного пространства во временное применяется обратное преобразование Фурье.

Этап 4: Вычисление фазы ряда. С помощью формализма аналитического сигнала вычисляется фаза ряда в виде аргумента отношения действительной части к мнимой части. Полученная фаза на данном этапе представляет собой главные значения функции arctg в пределах годичного периода.

Этап 5: Вычисление непрерывной функции фазы на всем интервале наблюдения.

Этап 6: Вычитание линейной составляющей. Из полученной непрерывной функции фазы вычитается линейная составляющая и получаются фазовые флуктуации данного температурного ряда.

Для того, чтобы сгруппировать станции со схожим климатом был разработан алгоритм сравнения их фаз путем расчета попарного коэффициента корреляции для каждой из фаз, полученных ранее, получении типовых фаз на их основе и конечной группировки станций по их фазам.

На рисунке 5 приведена обобщенная схема расчета типовых фаз.

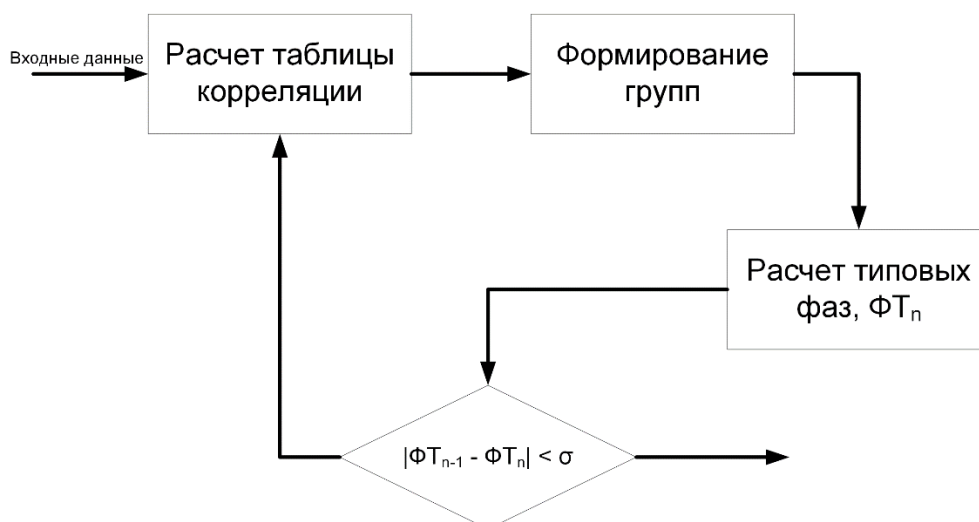


Рис. 5. Обобщенная схема расчета типовых фаз

Этап 1: Вычисленные фазы для каждой из групп подаются на вход узла расчета таблица корреляции, где высчитывается попарный коэффициент корреляции.

Этап 2: Полученная таблица корреляции фаз передается в узел формирования групп. Здесь формируются группы из тех температурных рядов, которые имеют с формирующим группу рядом коэффициенты корреляции выше заданного уровня.

Этап 3: На основе сформированных групп в узле расчета типовых фаз рассчитываются типовые фазы текущей итерации как среднеарифметическая оценка первого уровня.

Описанные процедуры включались в итеративный процесс, в котором входными данными были типовые фазы предыдущего уровня, а выходными – типовые фазы следующего уровня для каждого температурного ряда, и, соответственно, для каждой метеостанции. При этом, группы изменялись в соответствии с заданным пороговым критерием, главным образом за счет включения элементов, не входивших в данную группу на предыдущем шаге. Задаваемый порог определяет размер группы и связность исходных фаз с текущей в процессе итераций.

Все вычисления алгоритма делятся на две части: подготовительная часть – расчет фазы температурных рядов, основная часть – фазовая группировка.

Рассмотрим возможность параллельного вычисления этих блоков.

В случае подготовительной части, так как расчет фазы происходит линейно, то есть следующий этап выполняется на основе результатов предыдущего этапа (рис. 4), параллельное вычисление внутри алгоритма невозможно. Однако, исходя из того, что должно обрабатываться множество температурных рядов (от трехсот и более) целесообразным является параллельная подготовка каждого ряда.

Все этапы параллельных вычислений основываются на модели MapReduce, представленной на рисунке 6.

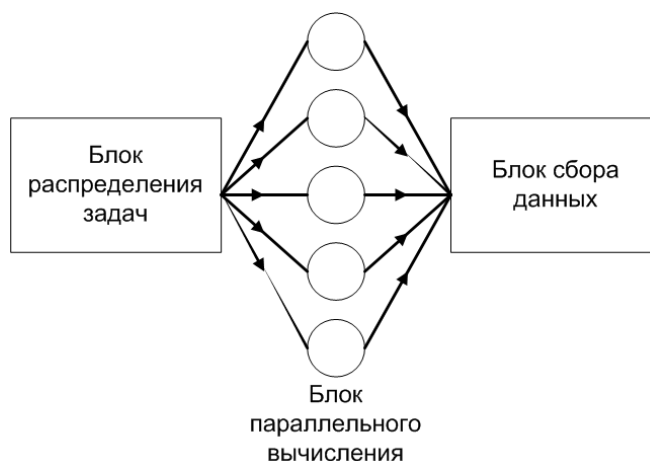


Рис. 6. Обобщенная схема параллельного расчета

Таким образом, блок распределения задач декомпозирует входные данные по доступным мощностям для параллельного расчета фазы температурных рядов (процедура Map), после чего полученные результаты на каждом узле собираются в блоке сбора данных и подготавливаются для дальнейшего анализа (процедура Reduce).

Для параллельного расчета фазовой группировки – так как этот процесс (рис 5.) итеративен, то на каждой итерации используются данные, полученные на предыдущем этапе, а это значит, что параллельное выполнение всей части невозможно.

На этапе фазовой группировки представляется возможным параллельное вычисление внутри алгоритма. На блок-схеме фазовой группировки этапы, подходящие для распределения выделены цветом (рис. 7), в каждом таком блоке подразумевается использованием общей схемы параллельного расчета (рис. 6).

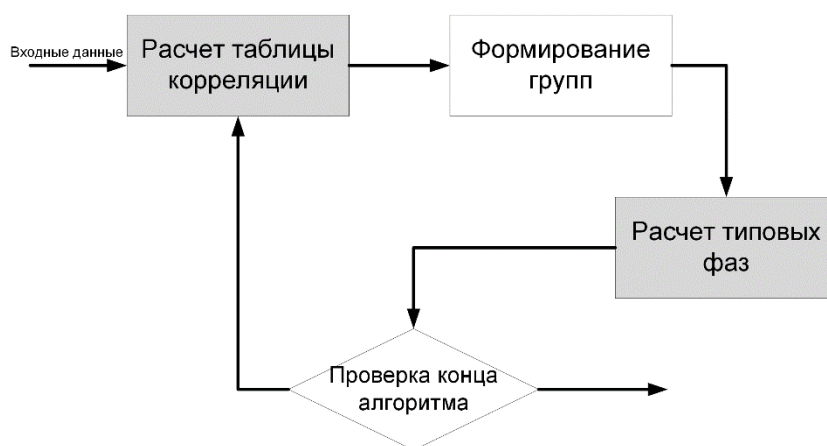


Рис. 7. Распараллеливание этапа расчета фазовой группировки

Таким образом, на каждой итерации расчет таблицы корреляции может быть распределен между доступными вычислительными мощностями, затем, после сбора полученных данных в единую таблицу, на основе анализа итоговой таблицы возможно параллельное вычисление типовых фаз.

4. Технологическая схема реализации и развертывания многомашинной системы для фазовой кластеризации

4.1. Описание алгоритма кластеризации метеорологических станций по фазе температурного ряда

Основным источником энергии, обуславливающим природно-климатические процессы на Земле, является Солнце. Энергия поступает в виде солнечной радиации, несущей свет и тепло в разных диапазонах длин волн. Прямое и рассеянное излучение поглощается поверхностью Земли, которая нагревается, и сама становится источником тепла. С учетом географической широты самое большое количество тепла получают нижние слои атмосферы, непосредственно прилегающие к земной поверхности, они и нагреваются до наиболее высоких температур. Таким образом, земная радиация определяет температурный режим и соответствующие ему циркуляции в атмосфере. Температура является первичным фактором формирования погоды и климата [29].

Анализ структуры климатических полей (районирование, классификация и т.п.) направлен на выявление закономерностей формирования различных типов климата в глобальной климатической системе. С другой стороны, определение территориальных границ типов климата, различных по своим свойствам, позволяет организовывать накопленные к настоящему времени значительные объемы климатических данных в небольшое число информационных структур. Предложен оригинальный метод выделения климатических областей, позволяющий решить обе задачи в любых пространственных и временных масштабах [30].

Среднемесячные изменения за исследуемый период формируют колебательный процесс с квазипериодом в один год. Характерными признаками, определяющими погоду, являются температурные изменения, отклоняющиеся от годового цикла. Однако их непротиворечивое выделение затруднено [31].

Частотный спектр температурного колебательного процесса является узкополосным и для большинства метеостанций имеет одну моду, что позволило использовать условия причинности и ввести фазу колебания, применяя теория аналитического сигнала [32, 33]. Годовая составляющая фазы представляет собой линейную функцию, она удалялась из фазы методом наименьших квадратов на интервале 55 лет (рис. 8.). Оставшиеся флуктуации фазы позволяют непротиворечиво исследовать их согласованное поведение или синхронность, применяя в виде этой меры коэффициент корреляции по Пирсону.

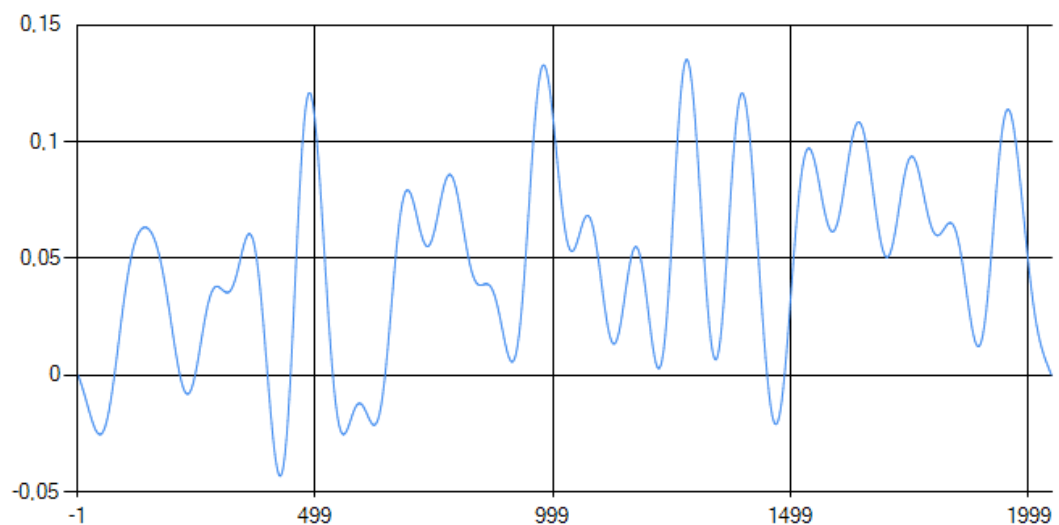


Рисунок 8. Линейная флуктуация фазы

4.1.1. Загрузка входных данных

Входные данные представляют собой текстовый файл, каждая строка которого представляет собой среднемесячные температуры определенного месяца для данного количества метеорологических станций.

Первым этапом алгоритма является загрузка входных среднемесячных температур (исследования проводились на данных за 55 лет 343 станций).

Исходный код метода загрузки входных данных:

```
public static ArrayList<ArrayList<Double>> ReadFromFile(File file){
    ArrayList<ArrayList<Double>> finalTemp = new ArrayList<ArrayList<Double>>();
    try {
        BufferedReader reader = new BufferedReader(new FileReader(file));
        String line;
        List<String> lines = new ArrayList<String>();
```

```

        int counter = 0;
        while((line = reader.readLine()) != null){
            counter++;
            String[] parts = line.split(" ");
            lines.add(line);
        }
        boolean flag = false;
        for (int i = 0; i < lines.size(); i++){
            String[] parts = lines.get(i).split("\t");
            counter = 0;
            for (int j = 0; j < parts.length; j++){
                if (!parts[j].equals("")){
                    if (!flag) finalTemp.add(new ArrayList<Double>());
                    finalTemp.get(counter).add(Double.parseDouble(parts[j]));
                    counter++;
                }
            }
            flag = true;
            counter=0;
        }

    } catch (FileNotFoundException ex) {
        Logger.getLogger(FileManager.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IOException ex) {
        Logger.getLogger(FileManager.class.getName()).log(Level.SEVERE, null, ex);
    }

    return finalTemp;
}

```

4.1.2. Вычисление фазы температурного ряда

В общем виде блок-схема алгоритма вычисления фазы представлена на рисунке 9.

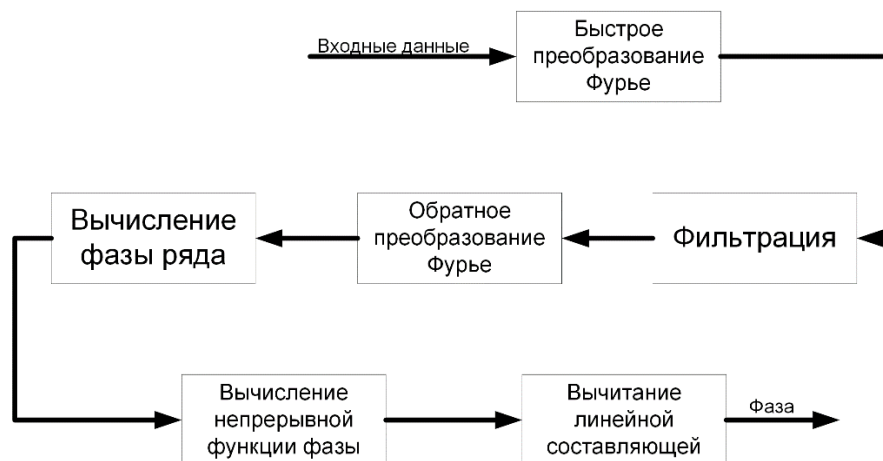


Рисунок 9. Общая схема алгоритма расчета фазы температурного ряда

С вычислительной точки зрения аналитический сигнал, соответствующий данной действительной функции T получается путем обнуления одной половины ее Фурье-преобразования и последующего обратного Фурье-преобразования.

После загрузки входных данных температурный ряд каждой станции преобразуется в сумму косинусоидальных и синусоидальных функций с помощью Дискретного Преобразования Фурье (ДПФ) (1).

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}, k = 0 \dots N - 1 \quad (1)$$

Дискретное преобразование Фурье реализовано на основе алгоритма Быстрого Преобразования Фурье (БПФ). Алгоритм БПФ позволяет существенно уменьшить время вычисления N -точечного ДПФ лишь при условии, что число N имеет много сомножителей. Однако существуют и другие эффективные алгоритмы расчета ДПФ последовательностей, также требующие выполнения порядка $N \log N$ операций. К ним относится алгоритм Блестейна, применяемый при любых N и основанный на цифровой фильтрации, эквивалентной вычислению ДПФ [34].

Исходный код метода, реализующего алгоритм Блестейна:

```

public static void transformBluestein(double[] real, double[] imag) {
    if (real.length != imag.length)
        throw new IllegalArgumentException("Mismatched lengths");
    int n = real.length;
    if (n >= 0x20000000)

```



```

        throw new IllegalArgumentException("Array too large");
    int m = Integer.highestOneBit(n * 2 + 1) << 1;

    double[] cosTable = new double[n];
    double[] sinTable = new double[n];
    for (int i = 0; i < n; i++) {
        int j = (int)((long)i * i % (n * 2)); // This is more accurate than
j = i * i

        cosTable[i] = Math.cos(Math.PI * j / n);
        sinTable[i] = Math.sin(Math.PI * j / n);
    }

    double[] areal = new double[m];
    double[] aimag = new double[m];
    for (int i = 0; i < n; i++) {
        areal[i] = real[i] * cosTable[i] + imag[i] * sinTable[i];
        aimag[i] = -real[i] * sinTable[i] + imag[i] * cosTable[i];
    }

    double[] breal = new double[m];
    double[] bimag = new double[m];
    breal[0] = cosTable[0];
    bimag[0] = sinTable[0];
    for (int i = 1; i < n; i++) {
        breal[i] = breal[m - i] = cosTable[i];
        bimag[i] = bimag[m - i] = sinTable[i];
    }

    double[] creal = new double[m];
    double[] cimag = new double[m];
    convolve(areal, aimag, breal, bimag, creal, cimag);

    for (int i = 0; i < n; i++) {
        real[i] = creal[i] * cosTable[i] + cimag[i] * sinTable[i];
        imag[i] = -creal[i] * sinTable[i] + cimag[i] * cosTable[i];
    }
}

```

Для подавления шума используется медианная фильтрация. Одномерный медианный фильтр основан на эвристическом методе обработки и представляет собой скользящее окно, охватывающее нечетное число элементов.

Далее, для возвращения из частотного пространства во временное применяются обратное преобразование Фурье (2).

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn}, n = 0 \dots N - 1 \quad (2)$$

Введение аналитического сигнала позволяет, с помощью формализма аналитического сигнала, определить фазу как арктангенс отношения его мнимой части к действительной. Полученная фаза на данном этапе представляет собой главные значения функции \arctg в пределах годового периода (рис. 10).

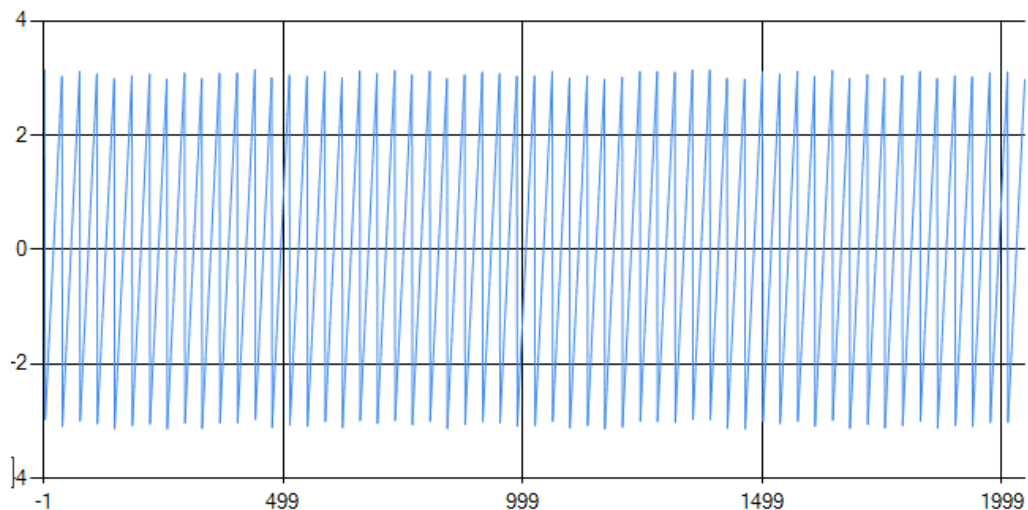


Рисунок 10. Главные значения функции \arctg в пределах годового периода для входного температурного ряда

Для дальнейшего расчета вычисляется непрерывная функция фазы на всем интервале наблюдения, затем из нее вычитается линейная составляющая, после чего получатся фазовые флуктуации, представленные на рисунке 1.

Исходный код вычисления непрерывной функции (PhaseLinking) и вычитания линейной составляющей (RemoveLinear):

```
public void PhaseLinking(){
    double c = 0;
    double d = phase[1] - phase[0];
    double n = phase.length;
    for (int i = 1; i < n; i++){
        double dd = d;
        if (i < n - 1) d = phase[i+1] - phase[i];
        double cc = c;
        if (dd > Math.PI) cc = c - 2 * Math.PI;
        if (-dd > Math.PI) cc = c + 2 * Math.PI;
        phase[i] = (phase[i] + cc);
        c = cc;
    }
}
```

```

    }
}

public void RemoveLinear(){
    finals = new double[phase.length];
    double k = (phase[phase.length - 1] - phase[0]) / (phase.length - 1);
    for (int i = 1; i <= phase.length; i++){
        finals[i-1] = phase[i-1] - k*(i-1) - phase[0];
    }
}
}

```

4.1.3. Алгоритм кластеризации

Общая блок-схема работы алгоритма кластеризации представлена на рисунке 11.

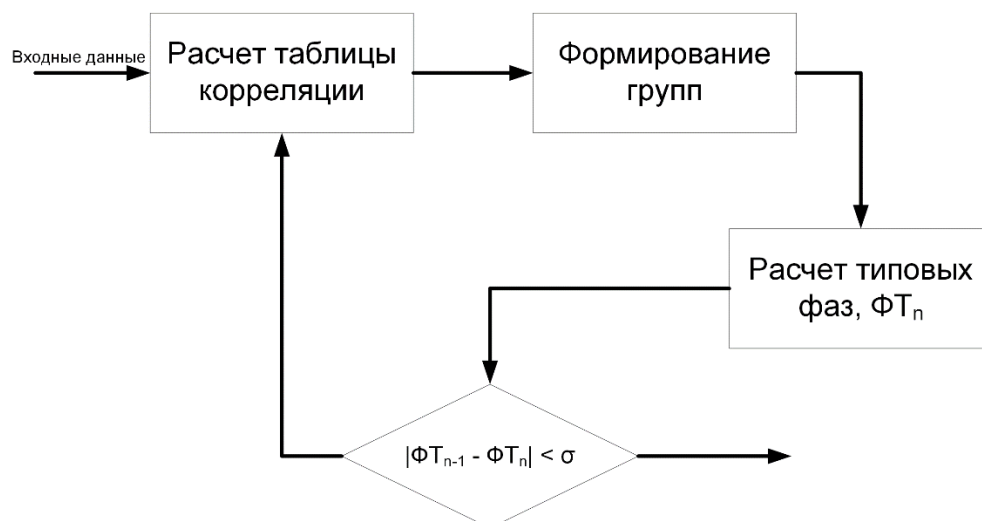


Рисунок 11. Общая схема работы алгоритма

Для выделения групп станций со схожим климатом реализован алгоритм, основанный на выделении типовых фаз, соответствующих основным группам с помощью вычисления попарной таблицы корреляции.

На первом этапе алгоритма вычисленные фазы подаются на вход узла расчета таблицы корреляции, где высчитывается попарный коэффициент корреляции для всех станций. Использовался коэффициент корреляции Пирсона (3).

$$r_{xy} = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}} \quad (3)$$

Исходный код расчета:

```
private double correlationCalc() {
    double averagel = 0, average2 = 0;
    int num = firstStantion.length;
    for (int i = 0; i < num; i++){
        averagel += firstStantion[i];
        average2 += secondStantion[i];
    }
    averagel = averagel / num;
    average2 = average2 / num;

    double part1 = 0;
    double part2 = 0;
    double part3 = 0;
    for (int i = 0; i < num; i++){
        double dif1 = firstStantion[i] - averagel;
        double dif2 = secondStantion[i] - average2;
        part1 += dif1*dif2;
        part2 += dif1*dif1;
        part3 += dif2*dif2;
    }
    double part4 = Math.sqrt(part2 * part3);
    return part1 / part4;
}
```

После расчета таблицы корреляции формируются группы метеостанций, фазы температурных рядов которых имеют коэффициенты корреляции с фазами других метеостанций выше заданного порога r . В каждой группе определялась своя текущая фаза как среднее арифметическое фаз, попавших в группу. Для сгруппированных станций исходные фазовые функции заменяются на текущие и процесс повторяется итеративно, при этом группы изменяются в соответствии с пороговым критерием r за счет включения элементов, не входивших в данную группу на предыдущем шаге. Как результат сходимости процесса к неподвижной точке состав групп стабилизировался, текущие фазы перестают меняться в связи с формированием их типовых свойств.

Исходный код формирования типовых групп и вычисления типовых фаз для станций:

```
private void allocateGroups() {
```

```

groups = new ArrayList[corrTable.size() + 1];
for (int i = 0; i < groups.length; i++)
    groups[i] = new ArrayList<Integer>();
int i = 0;
int j = 0;
try{
for (i = 0; i < groups.length; i++){
    for (j = 0; j < groups.length; j++){
        if (i == j) groups[i].add(j);
        else{
            if (j > i){
                if (corrTable.get(i).get(j-i-1) >= g) groups[i].add(j);
            }
            else{
                if (corrTable.get(j).get(i-j-1) >= g) groups[i].add(j);
            }
        }
    }
}
} catch (Exception e){
    System.out.println(e.toString());
    System.out.println(i + " " + j + " " + corrTable.size() + " " +
corrTable.get(i).size());
}
}

private void calcTypical(){

    for (int i = 0; i < phases.get(0).size(); i++){
        typical.add(phases.get(0).get(i));
    }
    for (int i = 1; i < members.size(); i++){
        for (int j = 0; j < phases.get(i).size(); j++){
            typical.set(i, typical.get(j) + phases.get(i).get(j));
        }
    }
    for (int i = 0; i < typical.size(); i++){
        typical.set(i, typical.get(i) / members.size());
    }

}

```

После формирования типовых фаз из списка групп удаляются повторяющиеся типовые, состоящие из одних и тех же групп, после чего рассчитывается таблица корреляции исходных фаз с вычисленными типовыми.

Итог работы алгоритма представлен на рисунках 12-13.



Рисунок 12. Результат работы алгоритма

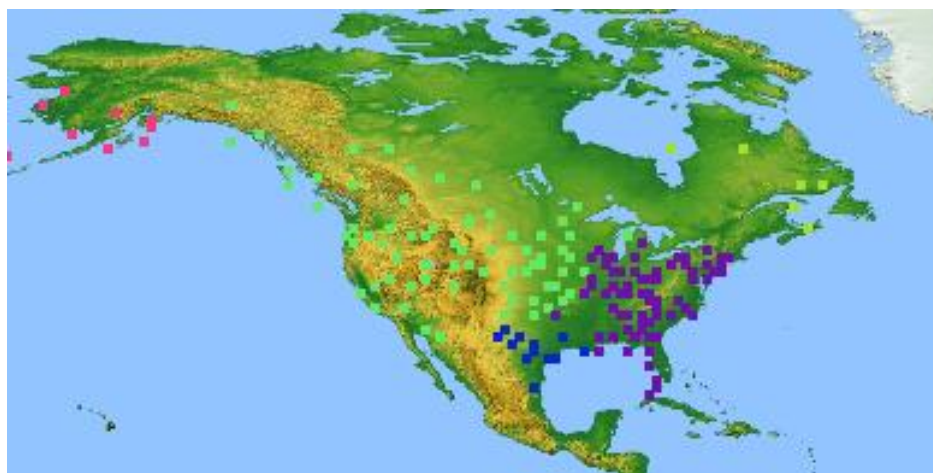


Рисунок 13. Результат работы алгоритма.

4.2. Описание грид системы

Для проведения параллельных вычислений на независимых вычислительных узлах была разработана вычислительная среда, которая представляет собой горизонтально масштабируемый кластер распределенных в пространстве серверов исследователей со своими вычислительными узлами-агентами обработки.

Разработанная грид система состоит из двух основных частей: центр управления, отвечающий за организацию всех вычислительных процессов в

среде, и клиентское приложение, отвечающее за выполнение задач, полученных от центра управления.

4.2.1. Центр управления

Центр управления представляет собой совокупность графического (оконного) интерфейса, с помощью элементов которого можно запустить сервер, просматривать доступные узлы вычислений, а также запустить задачу, и консольного интерфейса, с помощью которого можно сконфигурировать задачу для выполнения в распределенной среде. В консольный вывод также поступают все сервисные сообщения о текущей задаче. Графический интерфейс центра управления представлен на рисунке 14.

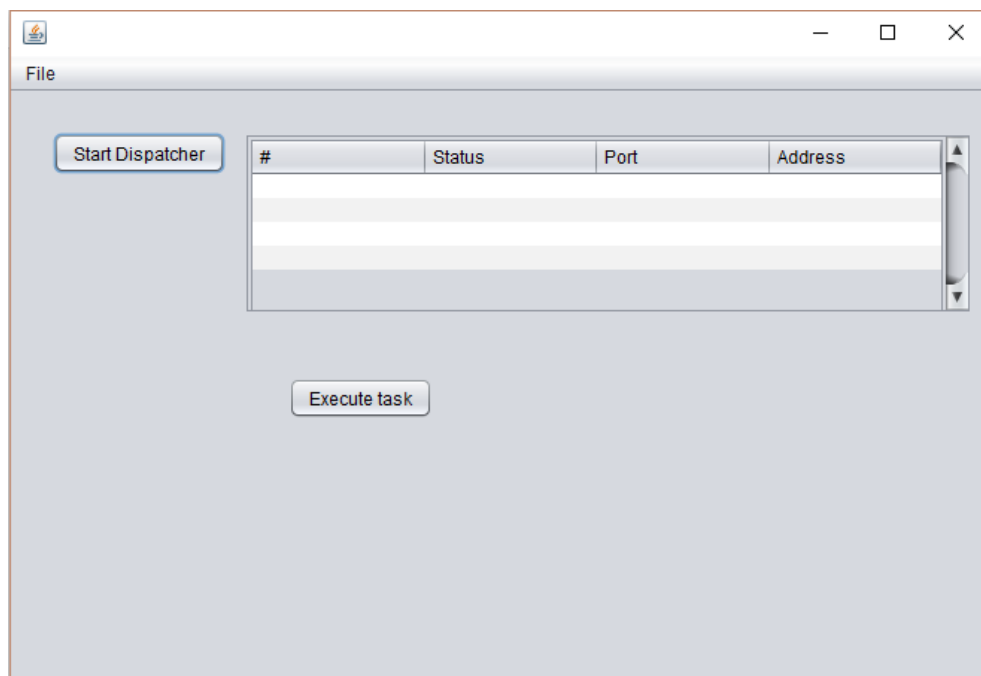


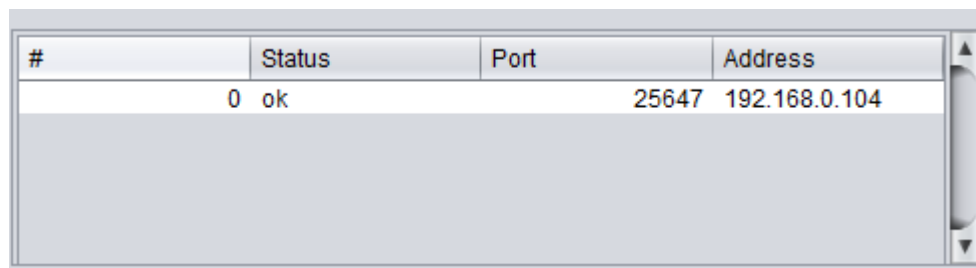
Рисунок 14. Графический интерфейс центра управления грид системой

4.2.1.1. Начало работы

Для начала работы в грид системе необходимо запустить менеджер подключений (Connection Manager) с помощью кнопки Start Dispatcher в графическом интерфейсе центра управления.

После начала своей работе менеджер подключений постоянно прослушивает порт, на который вычислительные узлы посылают служебное сообщение с информацией о начале своей работе (IP адрес и порт). Если такое

сообщение поступило, менеджер заносит подключенный узел в список узлов, который отображается в таблице графического интерфейса (рис. 15).



#	Status	Port	Address
0	ok	25647	192.168.0.104

Рисунок 15. Список подключенных вычислительных узлов

Для хранения информации о подключенных узлах используется список элементов класса `Node`, реализующий методы отправки специальных сообщений (проверки состояния, начала передачи файла) и состоящих из следующих характеристик: прослушивающий порт, порт файлового трансфера, IP-адрес и статус.

```
public class Node {  
    public int port;  
    public int transferPort;  
    public String address;  
    public String status;  
}
```

После того, как в списке появился подключенный узел менеджер начинает постоянный мониторинг статуса узла. На сообщенный узлом адрес и порт посылается сообщение с запросом о статусе узла, если ответ не получен – узел считается отключенным и удаляется из списка, если ответ получен, то статус меняется на полученный. Возможно два варианта статуса узла – «ок» когда узел подключен и готов получить задачу на выполнение, «busy» – узел занят вычислением задачи.

Исходный код мониторинга подключенных узлов:

```
@Override  
public void run() {  
    while (works) {  
        ArrayList<Integer> disconnected = new ArrayList<Integer>();  
        for (Node n : nodes) {  
            try (Socket socket = new Socket(InetAddress.getByName(n.address), n.port)) {
```



```

        System.out.println("Asking node number " + nodes.indexOf(n));
        InputStream sin = socket.getInputStream();
        DataInputStream in = new DataInputStream(sin);
        OutputStream sout = socket.getOutputStream();
        DataOutputStream out = new DataOutputStream(sout);

        out.writeUTF("status");
        out.flush();
        String line = in.readUTF();
        n.status = line;
        refreshNodesTable();
    } catch (Exception x) {
        System.out.println("Error while asking for node's status " + x);
        disconnected.add(nodes.indexOf(n));
    }
}

for (int i = disconnected.size() - 1; i >= 0; i--) {
    int index = disconnected.get(i);
    nodes.remove(index);
    System.out.println("Node #" + index + " disconnected");
    refreshNodesTable();
}

try {
    Thread.sleep(5000);
} catch (InterruptedException ex) {
    Logger.getLogger(ConnectionManager.class.getName()).log(Level.SEVERE,
null, ex);
}
}
}

```

4.2.1.2. Создание вычислительной задачи

В процессе подготовки задачи к вычислению с помощью грид системы пользователь формирует карту задач в виде графа, на которой указываются задачи и их зависимости между собой. Так, пользователь указывает связи тех задач, которым необходимо дождаться выполнения предыдущего этапа вычислений. Все задачи хранятся в виде списка элементов класса Task состоящих из следующих характеристик: имя java-файла для выполнения, имя главного метода, тип задачи, имя класса и id-номер задачи.

```

public class Task {
    public String javaFileName;

```

```

public String startingMethodName;
public String type;
public String className;
public int nodeNumber;
}

```

Существует три типа вычислительных задач:

- Вычислительная задача – задача, которая распределяется для вычисления по подключенным узлам.
- Файловая задача – задача, которая собирает полученные данные от узлов.
- Глобальная файловая задача – задача, которая подготавливает входные данные для распределения по подключенным узлам.

Для хранения всего графа задач создается класс Matrix, которая хранит матрицу связей всех узлов, а также список всех задач. Данный класс реализует метод мастера создания:

```

public TaskMatrix(){
    System.out.println("Task matrix forming process...");
    System.out.println("Enter number of nodes:");
    Scanner sc = new Scanner(System.in);
    int dimension = sc.nextInt();
    matrix = new int[dimension][dimension];
    tasks = new Task[dimension];
    for (int i = 0; i < dimension; i++){
        for (int j = 0; j < dimension; j++){
            matrix[i][j] = 0;
        }
    }
    for (int i = 0; i < dimension; i++){
        System.out.println("Forming node #" + (i+1));
        System.out.println("Enter type of node. If it's calculation node - enter \"calc\");
        System.out.println("If it's data input forming node - enter \"data\");
        System.out.println("If it's global data merging node - enter \"global\");
        String type = sc.next();
        System.out.println("Enter connections, when done enter \"!\");
        while(sc.hasNextInt()){
            int temp = sc.nextInt();
            matrix[i][temp-1] = 1;
        }
    }
}

```

```

        sc.next();
        System.out.println("Enter java file name");
        String javaFile = sc.next();
        System.out.println("Enter class name");
        String className = sc.next();
        System.out.println("Enter method name to execute");
        String methodName = sc.next();
        tasks[i] = new Task(javaFile, methodName, type, className, i);
    }
    System.out.println("Forming done");
    System.out.println("Please put all java files to \"javaFiles\" folder");
}

```

Выстраивая диалог с пользователем, программа полностью формирует список задач и матрицу связей между ними. Также класс `Matrix` реализует метод, возвращающий задачи, которые могут быть выполнены на текущий момент, в процессе выполнения задачи в матрице помечаются уже выполненные узлы.

Все указанные java-файлы для каждого узла должны быть помещены в папку «`javaFiles`», а входные данные для первой задачи помещаются в папку «`localInput`».

4.2.1.3. Запуск выполнения задачи

После того, как сервер запущен, сформирована вычислительная задача и подготовлены все файлы для вычисления пользователь может начать выполнение задачи с помощью кнопки «`Execute task`» (рис. 7).

После начала выполнения задачи запускается менеджер задачи (`Execute Manager`) который итеративно получает список доступных для выполнения задач, после чего, в зависимости от типа задачи:

- Глобальная файловая задача – запускает на выполнение связанный с задачей java-файл локально, назначая ей папку входных данных – `localInput` (папка общих входных данных), папкой выходных данных – `inputData` (папка входных данных для вычислительной задачи).

- Вычислительная задача – получает список файлов в папке inputData, после чего каждый файл отправляется свободному для вычислений узлу вместе со связанным java-файлом и назначается на выполнение. Параллельно создается файловый поток, который ожидает ответа от узла с выходными данными, которые помещаются в папку outputData в подпапке с номером узла (выходные данные, полученные с узлов).
- Файловая задача – запускает на выполнение связанный с задачей java-файл локально, назначая ей папку входных данных – outputData в подпапке с номером предыдущей вычислительной задачи, папкой выходных данных – localInput, для запуска дальнейшей итерации распределения по узлам.

Исходный код работы менеджера задач:

```
@Override
public void run() {
    boolean taskDone = matrix.getTaskDone();
    while (!taskDone) {
        ArrayList<Task> tasks = matrix.getTaskQue();
        for (Task t:tasks){
            if (t.type.equals("calc")){
                File input = new File(System.getProperty("user.dir") +
"\\inputData\\");
                File[] files = input.listFiles();
                for (File f:files){
                    Node n = null;
                    while (n == null ){
                        n = connectionManager.getFreeNode();
                    }
                    distributionManager.distributeFile(f, n, "input");
                    File jFile = new File(System.getProperty("user.dir") +
"\\javaFiles\\" + t.javaFileName);
                    distributionManager.distributeFile(jFile, n, "javaInput");
                    ClassExecuteThread exThread = new ClassExecuteThread(n,
t.className, t.startingMethodName, t.nodeNumber, t.nodeNumber);
                    exThread.start();
                    //classExecute(n, t.className, t.startingMethodName,
t.nodeNumber);

                    n.status = "busy";
                    connectionManager.refreshNodesTable();
                }
            }
        }
    }
}
```

```

        }
        else if(t.type.equals("data")){
            File jFile = new File(System.getProperty("user.dir") +
"\\javaFiles\\" + t.javaFileName);
            String classPath = System.getProperty("user.dir") + "\\javaFiles\\";
            String inputPath = System.getProperty("user.dir") + "\\outputData\\"
+ t.nodeNumber + "\\";
            String outputPath = System.getProperty("user.dir") +
"\\localInput\\";
            LocalJavaExecution.executeJava(jFile, t.className,
t.startingMethodName, classPath, inputPath, outputPath);
        }
        else if(t.type.equals("global")){
            File jFile = new File(System.getProperty("user.dir") +
"\\javaFiles\\" + t.javaFileName);
            String classPath = System.getProperty("user.dir") + "\\javaFiles\\";
            String inputPath = System.getProperty("user.dir") + "\\localInput\\";
            String outputPath = System.getProperty("user.dir") + "\\inputData\\";
            LocalJavaExecution.executeJava(jFile, t.className,
t.startingMethodName, classPath, inputPath, outputPath);
        }
    }
    boolean turnDone = false;
    while(!turnDone){
        int counter = 0;
        for (Task t:tasks){
            if (t.type.equals("calc")){
                File f = new File(System.getProperty("user.dir") +
"\\nodesData\\" + t.nodeNumber + "\\");
                File input = new File(System.getProperty("user.dir") +
"\\outputData\\");

                int inFiles = input.listFiles().length;
                File files[] = f.listFiles();
                if (files.length == inFiles) counter++;
            }
            else{
                counter++;
            }
        }
        if (counter == tasks.size()) turnDone = true;
    }
    for (Task t:tasks){
        matrix.setTaskDone(t.nodeNumber);
    }
    taskDone = matrix.getTaskDone();
}
}

```

4.2.2. Клиентское приложение

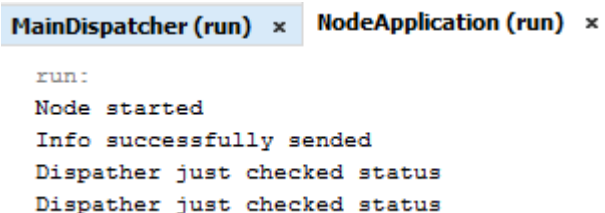
Клиентская часть выполнена в виде полностью консольного приложения. Никакого управления со стороны пользователя не подразумевается, он лишь предоставляет свои вычислительные ресурсы системе.

4.2.2.1. Начало работы

После запуска приложения пользователю необходимо ввести IP-адрес сервера центра управления, после чего приложение направит ему все необходимую информацию для дальнейшей работы и начнет прием команд. За отправку и прием всех сообщений отвечает менеджер соединения (Connection Manager):

```
public void run() {  
    try{  
        ServerSocket server = new ServerSocket(0);  
        transferSocket = new ServerSocket(0);  
        System.out.println("Node started");  
        port = server.getLocalPort();  
        address = InetAddress.getLocalHost().getHostAddress();  
        sendInfo(6758, "192.168.0.105");  
        status = "ok";  
        while(true){ getMessage(server); }  
    } catch (Exception x) {  
        System.out.println(x);  
    }  
}
```

Все события, происходящие на клиентской машине, отражаются в консоли в виде системных сообщений (рис. 16).



```
MainDispatcher (run) x NodeApplication (run) x  
run:  
Node started  
Info successfully sended  
Dispathther just checked status  
Dispathther just checked status
```

Рисунок 16. Консоль клиентского приложения

Клиентское приложение может получить от сервера несколько команд:

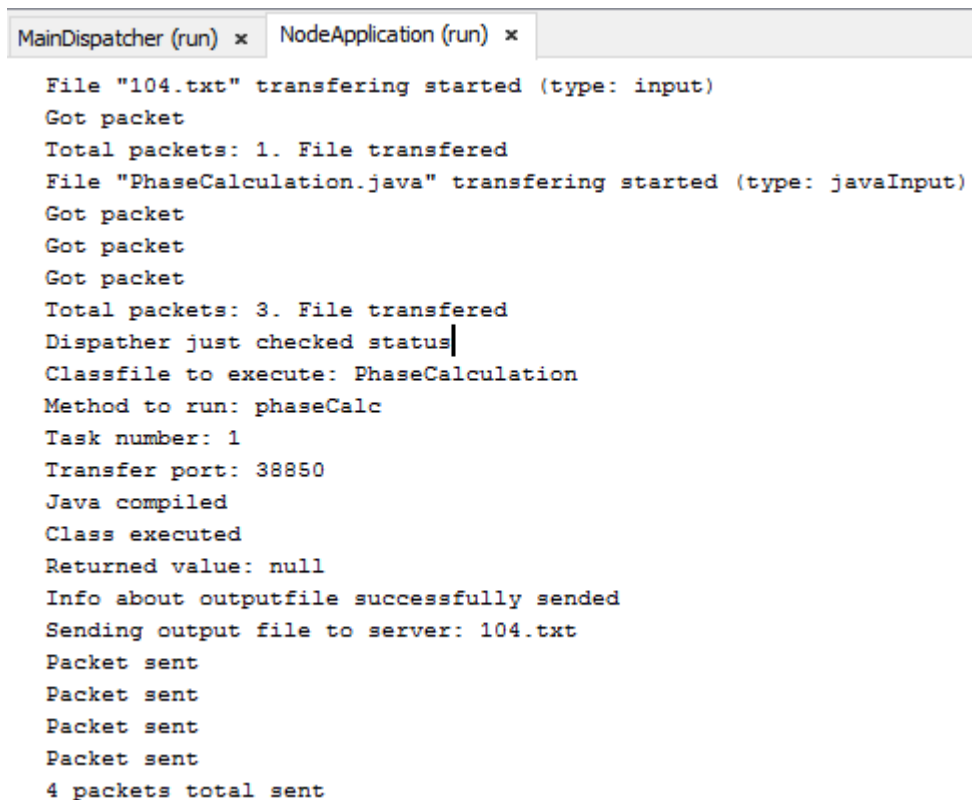
- Status – проверка статуса узла, клиент может ответить «ok» в случае если он свободен для выполнения задачи, «busy» если он занят выполнением задачи. Если клиент не отвечает на запрос статуса, сервер считает клиента утерянным.
- fileSending – старт отправки файла. Команду сервер сопровождает информацией о типе файла и его названии.
- classExecute – старт выполнения задачи. Сопровождается информацией об имени класса и необходимого метода, а также порт, на который необходимо направить полученный результат.

4.2.2.1. Выполнение задачи

Каждая направленная узлу задача состоит из следующих этапов:

1. Прием входных данных – клиенту направляется команда для приема данных с указанием типа данных как «входные данные» и соответствующим названием файла. Файл помещается в папку «inputData».
2. Прием java-файла – клиенту направляется команда для приема java-файла с указанием типа данных как «java-файл» и названием файла. Файл помещается в папку «javaInput».
3. Выполнение – клиенту направляется команда для выполнения направленной задачи с указанием класса для запуска и необходимого метода. Все выходные данные помещаются в папку «javaOutput».
4. Все файлы, хранящиеся в папке «javaOutput» передаются на указанный сервером порт.
5. Все служебные директории очищаются.

Консольный вывод всего процесса выполнения одной задачи представлен на рисунке 17.



```
MainDispatcher (run) x NodeApplication (run) x
File "104.txt" transferring started (type: input)
Got packet
Total packets: 1. File transfered
File "PhaseCalculation.java" transferring started (type: javaInput)
Got packet
Got packet
Got packet
Total packets: 3. File transfered
Dispatcher just checked status|
Classfile to execute: PhaseCalculation
Method to run: phaseCalc
Task number: 1
Transfer port: 38850
Java compiled
Class executed
Returned value: null
Info about outputfile successfully sended
Sending output file to server: 104.txt
Packet sent
Packet sent
Packet sent
Packet sent
4 packets total sent
```

Рисунок 17. Выполнение одной задачи

Описанные выше этапы развертывания программной среды для решения задач фазовой кластеризации температурных рядов по 343 станциям за 56 лет наблюдения были использованы при проведении программных экспериментов оценки быстродействия вычислений, описанных в главе 2.

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Данная работа посвящена разработке и исследованию итеративного параллельного алгоритма формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных, а также построению гирд-системы для выполнения данного алгоритма. Как и любой проект, вне зависимости от того какой характер он несет, научный или практический, работа имеет экономическую составляющую, которую необходимо оценить с целью предоставления полной картины значимости осуществленной работы и ее вклада в заданную предметную область.

Целью данного раздела является оценка и анализ экономических аспектов данной работы. Основными критериями для оценки служат эффективность проекта, затраты на его выполнение, а также перспективы его внедрения.

Для достижения поставленной цели необходимо решить следующие задачи:

- организация и планирование работ над проектом;
- расчет основных затрат и расходов на проект;
- оценка экономической эффективности проекта.

5.1. Организация и планирование работы

Одной из составляющих успешной реализации проекта служит рациональное планирование занятости каждого из его участников, а также определение сроков выполнения определенных этапов работы над проектом. В данном разделе приводится перечень этапов работы, исполнителей, а также оценивается степень участия каждого из участников в том или ином этапе. Данные по перечню работ и продолжительности работ представлены в таблице 3. Число исполнителей данного проекта равно двум – непосредственный исполнитель и научный руководитель.

Таблица 3 – Перечень работ и продолжительность их выполнения

Этапы работы	Испол	Загрузка
--------------	-------	----------

	нители	исполнителей
<i>постановка задачи</i>		
1. Постановка целей и задач, получение исходных данных	НР	НР - 100 %
<i>анализ</i>		
2. Выявлений требований к программному продукту	НР, И	НР – 100 %, И – 30%
3. Подбор и обзор литературы, обзор существующих решений	НР, И	НР – 30%, И – 100%
4. Разработка календарного плана	НР, И	НР – 100%, И – 10 %
<i>проектирование</i>		
5. Разработка модели системы	НР, И	НР – 80 % , И – 100%
6. Разработка алгоритма реализации	НР, И	НР – 70 %, И – 100%
<i>реализация</i>		
7. Разработка приложения	И	И – 100%
<i>тестирование</i>		
8. Тестирование	НР, И	НР – 10%, И – 100%
9. Анализ результатов	И	И – 100%
<i>документирование</i>		
10. Оформление пояснительной записки	И	И – 100%
11. Подведение итогов	НР, И	НР – 50%, И – 100%

5.1.1 Продолжительность этапов работ

В данном разделе осуществляется расчет продолжительности работ с использованием опытно-статистического метода. Данный метод имеет 2 способа реализации - аналоговый и экспертный. Так как аналоговый способ подразумевает наличие некоторого идентичной выполняемой научно-исследовательской работы (по всем значительным параметрам) проекта, то, в условиях отсутствия последнего, принято решение применять экспертный способ. Расчет продолжительности этапов работ приведен в таблице 4. Для построения таблицы используются следующие параметры:

- Ожидаемые (вероятные) значения продолжительности работ ($t_{ож}$):

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \text{ где}$$

t_{min} – минимальная продолжительность работы, дни;

t_{max} – максимальная продолжительность работы, дни;

- Продолжительность выполнения каждого этапа в рабочих днях ($T_{рд}$):

$$T_{РД} = \frac{t_{ож}}{K_{вн}} \cdot K_{д}, \text{ где}$$

$t_{ож}$ – продолжительность работы, дни;

$K_{вн}$ – коэффициент выполнения работ с учетом внешних факторов, влияющих на соблюдение предварительно определенных длительностей работы (в данном случае используется $K_{вн} = 1$);

$K_{д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ (принимает значения в интервале [1; 1,2], для расчетов используется величина 1,2 с целью максимально учесть непредвиденные временные затраты).

- Продолжительность выполнения этапа в календарных днях ($T_{КД}$):

$$T_{КД} = T_{РД} \cdot T_{К}, \text{ где}$$

$T_{К}$ – коэффициент календарности, служит для перехода от длительности работ в рабочих днях к их аналогам в календарных днях.

Данный коэффициент имеет следующую формулу:

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \text{ где}$$

$T_{КАЛ}$ – календарные дни (365 дней)

$T_{ВД}$ – выходные дни (52 или 104 дня, для шестидневной и пятидневной рабочей недели соответственно)

$T_{ПД}$ – праздничные дни (10 дней)

Таким образом, для шестидневной рабочей недели получаем следующий коэффициент календарности:

$$T_{К} = \frac{365}{365 - 52 - 10} = 1,205$$

Таблица 4 – Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.-дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9
Постановка целей и	НР	2	4	2,8	3,36	-	4,05	-

задач, получение исходных данных								
Выявлений требований к программному продукту	НР, И	2	3	2,4	2,88	0,86	3,47	1,04
Подбор и обзор литературы, обзор существующих решений	НР, И	8	12	9,6	3,46	11,52	4,17	13,88
Разработка календарного плана	НР, И	2	4	2,8	3,36	0,34	4,05	0,41
Разработка модели системы	НР, И	12	18	14,4	13,82	17,28	16,65	20,82
Разработка алгоритма реализации	НР, И	18	22	19,6	16,46	23,52	19,83	28,34
Разработка приложения	И	125	150	135	-	162	-	195,21
Тестирование	НР, И	4	6	4,8	0,58	5,76	0,7	6,94
Анализ результатов	И	4	6	4,8	-	5,76	-	6,94
Оформление пояснительной записки	И	4	6	4,8	-	5,76	-	6,94
Подведение итогов	НР, И	2	4	2,8	1,68	3,36	2,02	4,05
Итого				203,8	45,6	236,16	54,97	282,57

На рисунке 18 представлен линейный график осуществления проекта для исполнителя.

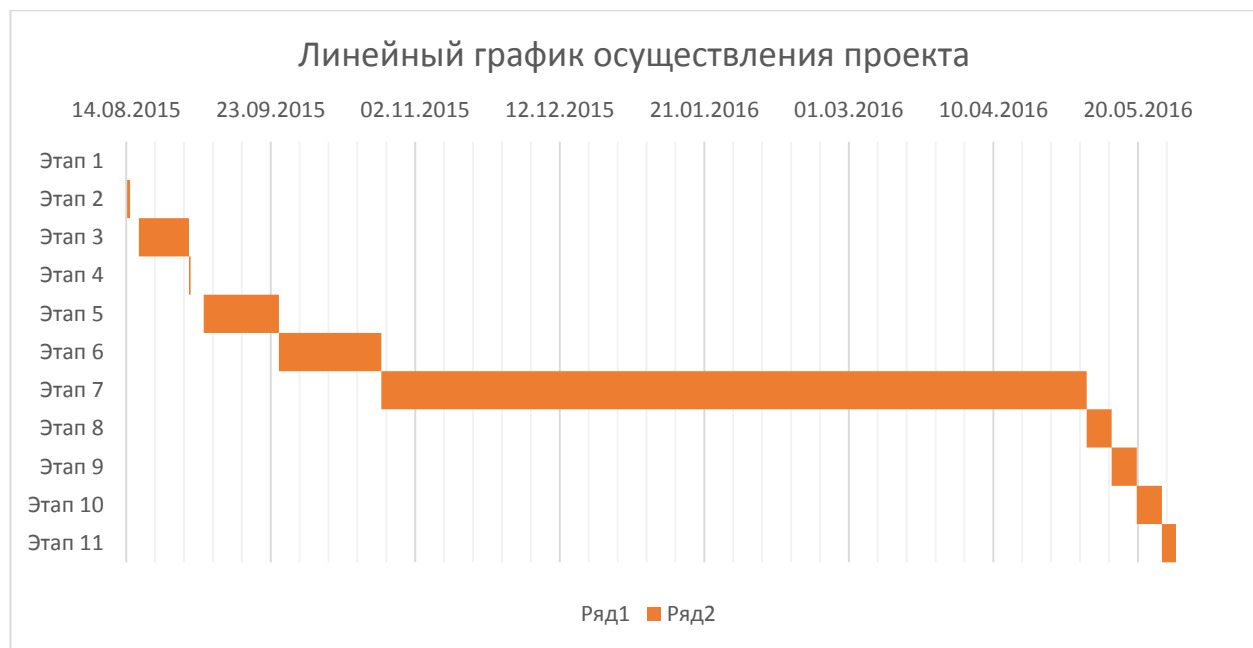


Рисунок 18. Линейный график осуществления проекта для исполнителя

5.1.2. Расчет накопления готовности проекта

В данном разделе производится оценка текущих состояний работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего (i-го) этапа выполнен общий объем работ по проекту в целом. Нарастание технической готовности работы и удельный вес каждого этапа представлены в таблице 5.

Для определения степени готовности проекта используется следующая формула:

$$CG_i = \frac{TP_i^H}{TP_{общ.}} = \frac{\sum_{k=1}^i TP_k}{TP_{общ.}} = \frac{\sum_{k=1}^i \sum_{j=1}^m TP_{km}}{\sum_{k=1}^I \sum_{j=1}^m TP_{km}}, \text{ где}$$

$TP_{общ.}$ – общая трудоемкость проекта;

$TP_i (TP_k)$ – трудоемкость i-го (k-го) этапа проекта;

TP_i^H – накопленная трудоемкость i-го этапа проекта по его завершении

$TP_{ij} (TP_{kj})$ – трудоемкость работ, выполняемых j-м участником на i-м этапе, здесь $j = \overline{1, m}$ – индекс исполнителя (в данном случае $m = 2$, так как в проекте 2 исполнителя). Расчет данной величины производится на основании столбцов 6 и 7.

Таблица 5 – Нарастание технической готовности работы и удельный вес каждого этапа

№ п/п	Этап	ТР _i , %	СГ _i , %
1	Постановка целей и задач, получение исходных данных	1,19	1,19
2	Выявление требований к программному продукту	1,33	2,52
3	Подбор и обзор литературы, обзор существующих решений	5,32	7,84
4	Разработка календарного плана	1,31	9,15
5	Разработка модели системы	11,04	20,19
6	Разработка алгоритма реализации	14,19	34,38
7	Разработка приложения	57,5	91,88
8	Тестирование	2,25	94,13
9	Анализ результатов	2,04	96,17
10	Оформление пояснительной записки	2,04	98,21
11	Подведение итогов	1,79	100

5.2. Расчет сметы затрат на выполнение проекта

С учетом специфики проделанной работы для рассматриваемого проекта производится оценка следующих расходов:

- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- прочие расходы.

Ввиду отсутствия привлечения помощи сторонних организаций, командировок, привлечения консалтинговых услуг, расходы по соответствующим статьям не оцениваются применительно к данной работе. Помимо вышеперечисленного, не рассматриваются также затраты на материалы ввиду их незначительности в масштабе данной работы.

5.2.1. Расчет заработной платы

В данном разделе расчет основной заработной платы производится на основе величины месячного оклада исполнителя и трудоемкости каждого этапа.

Расчет затрат на заработную плату представлен в таблице 6. Для расчета данной таблицы использовались следующие параметры:

- Месячный оклад исполнителей проекта (МО)

В данном случае рассматриваются следующие оклады:

- 1) научный руководитель: должность – доцент, степень – кандидат технических наук;
 - 2) исполнитель: младший научный сотрудник, степень – нет (оклад согласно месту прохождения преддипломной практики).
- Среднедневная тарифная заработная плата ($ЗП_{\text{дн-т}}$), рассчитывается по формуле:

$$ЗП_{\text{дн-т}} = \frac{МО}{24,83}, \text{ где}$$

МО – месячный оклад исполнителя,

Значение 24, 83 – количество рабочих дней при шестидневной рабочей неделе при условии 298 рабочих дней в году.

- Интегральный коэффициент ($K_{\text{и}}$)

– служит для перехода от базовой суммы заработной платы исполнителя, связанной с участием в проекте к полному заработку. Рассчитывается по формуле:

$$K_{\text{и}} = K_{\text{пр}} \cdot K_{\text{допЗП}} \cdot K_{\text{р}}, \text{ где}$$

$K_{\text{пр}}$ – премиальный коэффициент ($K_{\text{пр}} = 1,1$);

$K_{\text{допЗП}}$ – коэффициент дополнительной заработной платы ($K_{\text{допЗП}} = 1,188$);

$K_{\text{р}}$ – коэффициент районной надбавки ($K_{\text{р}} = 1,3$).

Таким образом, получаем следующий интегральный коэффициент:

$$K_{\text{и}} = 1,1 \cdot 1,188 \cdot 1,3 = 1,699$$

Таблица 6 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб. дни	Коэф-т	Фонд з/платы, руб.
НР	23 264, 86	936,97	46	1,699	73 227,95
И	14 874, 45	599,05	236	1,699	240 197,48

Итого:					313 425,43
---------------	--	--	--	--	------------

5.2.2. Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН) включают в себя следующие отчисления:

- пенсионный фонд;
- социальное страхование;
- медицинское страхование.

ЕСН составляет 30% от полной заработной платы по проекту:

$$C_{\text{соц}} = C_{\text{зп}} \cdot 0,3$$

Таким образом, для разрабатываемого проекта получаем:

$$C_{\text{соц}} = 313\,425,43 \cdot 0,3 = 94\,027,63 \text{ руб.}$$

5.2.3. Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, используемую оборудованием в ходе выполнения проекта и рассчитывается по формуле:

$$C_{\text{эл.об}} = P_{\text{об}} \cdot t_{\text{об}} \cdot C_{\text{э}}, \text{ где}$$

$P_{\text{об}}$ – мощность, потребляемая оборудованием, кВт;

$t_{\text{об}}$ – время работы оборудования, час;

$C_{\text{э}}$ – тариф на 1кВт·час.

Значения параметров

1) $C_{\text{э}}$ - для ТПУ составляет 5,257 руб/ кВт·час (с НДС)

2) $t_{\text{об}} = T_{\text{рд}} \cdot K_t$, где

- $T_{\text{рд}}$ – трудозатраты исполнителя из расчета на 8 часовой рабочий день;
- $K_t \leq 1$ – коэффициент использования оборудования по времени (в данном случае используется значение 0,9).

3) $P_{\text{об}} = P_{\text{ном}} \cdot K_c$, где

- $P_{\text{ном}}$ – номинальная мощность оборудования, кВт;
- $K_c \leq 1$ – коэффициент загрузки, зависит от средней степени использования номинальной мощности. Для данного проекта используется значение 1.

Расчет затрат электроэнергии представлен в таблице 7

Таблица 7 – Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{\text{ОБ}}$, час	Потребляемая мощность $P_{\text{ОБ}}$, кВт	Тариф $\text{Ц}_{\text{э}}$, руб/кВт·час	Затраты $\text{Э}_{\text{ОБ}}$, руб.
Персональный компьютер	$8*236*0,9 = 1699$	0,3	5,257	2 679,5
Персональный компьютер	$8*46*0,9 = 331$	0,3	5,257	552
Лазерный принтер	10	0,1	5,257	5,26
Итого				3 236,76

5.2.4. Расчет амортизационных расходов

Для расчета амортизационных расходов используется следующая формула:

$$C_{\text{АМ}} = \frac{N_{\text{А}} * \text{Ц}_{\text{ОБ}} * t_{\text{рф}} * n}{F_{\text{д}}}, \text{ где}$$

$N_{\text{А}}$ – годовая норма амортизации единицы оборудования;

$\text{Ц}_{\text{ОБ}}$ – балансовая стоимость единицы оборудования;

$t_{\text{рф}}$ – фактическое время работы оборудования в ходе выполнения проекта;

n – число единиц задействованного оборудования;

$F_{\text{д}}$ – действительный годовой фонд времени работы соответствующего оборудования.

Расчет амортизационных расходов представлен в таблице 8. Параметры расчета определяются следующим образом:

1) годовая норма амортизации единицы оборудования $N_{\text{А}}$

Величина $N_{\text{А}}$ определяется как обратное значение величины $С_{\text{А}}$, где $С_{\text{А}}$ – рамочные значения сроков амортизации $\equiv С_{\text{А}}$ согласно постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы».

2) балансовая стоимость единицы оборудования $\text{Ц}_{\text{ОБ}}$

Значение взято исходя из фактической стоимости ПК и принтера на момент разработки проекта.

- 3) фактическое время работы оборудования в ходе выполнения проекта $t_{рф}$
- 4) действительный годовой фонд времени работы соответствующего оборудования F_d .

Величина берется на основании фактического режима использования оборудования в текущем календарном году

- $t_{рф}$: для ПК – 2104 и 408, для принтера – 10

Таблица 8 - Амортизационные расходы

	H_A	$Ц_{об}$, руб.	$t_{рф}$, часов	n , шт	F_d , час	$C_{ам}$, руб
ПК	0,4	60 000	1699	1	2384	17 104
ПК	0,4	60 000	331	1	2384	3 332,2
принтер	0,5	15 000	10	1	500	150
Итого						20 586,2

5.2.5. Расчет прочих расходов

В данном разделе производится оценка расходов на выполнение проекта, которые не были учтены в предыдущих статьях. Величина прочих расходов составляет 10% от суммы всех предыдущих расходов. Таким образом:

$$C_{проч.} = (C_{зп} + C_{соц} + C_{эл.об.} + C_{ам}) \cdot 0,1$$

Для данного проекта получаем:

$$C_{проч.} = (313\,425,43 + 94\,027,63 + 3\,236,76 + 20\,586,2) \cdot 0,1 = 43\,127,6 \text{ руб.}$$

5.2.6. Расчет общей себестоимости разработки

Общая себестоимость разработки представляет суммарное значение затрат по всем статьям сметы затрат на разработку. Расчет общей себестоимости разработки представлен в таблице 9.

Таблица 9 - Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Основная заработная плата	$C_{зп}$	313 425,43
Отчисления в	$C_{соц}$	94 027,63

социальные фонды		
Расходы на электроэнергию	$C_{эл.}$	3 236,76
Амортизационные отчисления	$C_{ам}$	20 586,2
Прочие расходы	$C_{проч}$	43 127,6
Итого		474 403,62

5.2.7. Расчет прибыли

Одним из вариантов приблизительной оценки прибыли является определение ее размера в пределах от 5 % до 20% от полной себестоимости проекта. Приведем для данной проекта расчет минимальной и максимальной прибыли исходя из заданных пределов.

Таким образом:

$$\text{Максимальная прибыль} = 0,2 \cdot 474\,403,62 = 94\,880,72 \text{ руб.}$$

$$\text{Минимальная прибыль} = 0,05 \cdot 474\,403,62 = 23\,720,18 \text{ руб.}$$

5.2.8. Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли. Таким образом, для данного проекта получаем:

$$\text{НДС}_{\min} = (474\,403,62 + 23\,720,18) \cdot 18\% = 89\,662,28 \text{ руб.}$$

$$\text{НДС}_{\max} = (474\,403,62 + 94\,880,72) \cdot 18\% = 102\,471,18 \text{ руб.}$$

5.2.9. Цена разработки НИР

Данный параметр представляет сумму полной себестоимости, прибыли и НДС. Таким образом, получаем:

$$C_{\text{НИР_МИН}} = (474\,403,62 + 23\,720,18 + 89\,662,28) = 587\,786,08 \text{ руб.}$$

$$C_{\text{НИР_МАКС}} = (474\,403,62 + 94\,880,72 + 102\,471,18) = 671\,755,52 \text{ руб.}$$

5.3. Оценка экономической эффективности проекта

Экономический эффект от внедрения разработанного проекта в количественном отношении выразить сложно. Для этого требуется провести специальное трудоемкое исследование, что невозможно в рамках данной

работы, поскольку, во-первых, предполагает объём равный все дипломной работе, во-вторых, исполнитель не обладает компетенцией для выражения полного экономического эффекта в количественном отношении. Однако в качественном отношении внедрение разработанной системы окажет влияние на фундаментальные научные и инженерные задачи. Главный эффект заключается в формировании скрытых зависимостей, которые невозможно обнаружить без использования больших вычислений на основе огромного числа данных.

В современной метеорологии данные о погоде собираются более чем шестьдесят раз в секунду без перерывов. При этом собирается не только информация о температуре, но также, например, о влажности, направлении ветра, давлении и других параметрах. Только за сутки полученная информация может достигать нескольких гигабайт.

В конечном итоге за несколько лет по всему миру собирается огромное количество данных, проанализировать которые без использования специальных алгоритмов и вычислительных мощностей не представляется возможным. Одним из вариантов анализа таких данных является возможность формирования климатических областей на основе флуктуаций фазовых отклонений их температурных рядов. С помощью полученных таким образом данных можно наблюдать глобальные изменения климата, а также возможно предсказание таких изменений.

Общепризнанным является класс фундаментальных научных и инженерных проблем Grand Challenges, эффективное решение которых возможно только с использованием мощных вычислительных ресурсов. Одной из областей таких научных и инженерных проблем является предсказание погоды, климата и глобальных изменений в атмосфере. Таким образом проектирование распределенной грид-системы для вычисления описанного алгоритма окажет влияние на решение научных проблем из списка Grand Challenges.

Формирование климатических областей позволит получать карты их движения, что является более наглядной информацией, чем набор

разрозненных чисел, характеризующих среднюю температуру некоторой точки Земли. Анализ таких данных с помощью вычислительных мощностей поможет более наглядно изучить изменения климата.

Разработка распределенной грид-системы также позволит создавать кластеры, направленные на обработку больших данных. Например, понижение дискретизации данных с 60 раз в секунду до среднечасовых или среднесуточных, в зависимости от целей исследователя.

5.4. Оценка научно-технического уровня НИР

Научно-технический уровень характеризует влияние проекта на уровень и динамику обеспечения научно-технического прогресса в данной области. Для данной оценки используется метод балльных оценок, сущность которого заключается в присвоении каждому из показателей НИР определенного количества баллов по соответствующей для данного показателя шкале.

Научно-технический уровень определяется на основании его интегрального показателя, который выражается следующей формулой:

$$I_{НТУ} = \sum_{i=1}^3 R_i \cdot n_i, \text{ где}$$

$I_{НТУ}$ - интегральный индекс научно-технического уровня;

R_i – весовой коэффициент i -го признака научно-технического эффекта;

n_i – количественная оценка i -го признака научно-технического эффекта в баллах;

Оценка научно-технического уровня представлена в таблице 10.

Таблица 10 – Оценка научно технического уровня НИР

Значимость	Фактор НТУ	Уровень фактора	Выбранный балл	Обоснование выбранного балла
0,4	Уровень новизны	Новая	6	Применение фазовой группировки для метеоданных, разработка распределенной грид-системы
0,1	Теоретический уровень	Разработка	6	Разработка нового способа объединения

		способа		информации
0,5	Возможность реализации	В течение первых лет	10	Реализуется на основе хорошо известных технологий

Таким образом, для данного проекта получаем следующий показатель научно-технического уровня:

$$I_{\text{НТУ}} = 0,4 \cdot 7 + 0,1 \cdot 6 + 0,5 \cdot 10 = 2,4 + 0,6 + 5 = 8$$

В таблице 11 приводится оценка качественных уровней НИР.

Таблица 11 - Качественная оценка показателей НИР

Уровень НТЭ	Показатель НТЭ
Низкий	1-4
Средний	4-7
Высокий	8-10

На основании таблицы 1.9 данная работа относится к высокому уровню научно технического эффекта. Высокий уровень НТЭ обусловлен тем, что данная работа представляет принципиально новый подход в формировании скрытых структур для метеорологических данных.

Вывод

В данном разделе была произведена оценка различных экономических аспектов разработанного проекта. На основании полученных результатов проекту можно дать следующую характеристику:

по времени разработки (10 месяцев) проект относится к краткосрочному виду проектов;

по масштабности (с учетом себестоимости и времени разработки) проект можно отнести к категории малых проектов;

наличие новизны метода и актуальность решаемой проблемы повышают конкурентоспособность проекта в заданной предметной области.

6. Социальная ответственность

6.1. Введение

Разработанный в рамках магистерской диссертации проект является математическим и программным обеспечением, предназначенным для формирования скрытых функционально-детерминированных на основе анализа геофизических данных. Разработка программы велась исключительно при помощи компьютера. Пользователями данного программного обеспечения являются компании, занимающиеся анализом метеоданных. Независимо от конкретного применения, взаимодействие пользователя с разработанной программой в любом случае производится с помощью программных и аппаратных средств ПЭВМ, а также с помощью периферийных устройств, подключенных к ПЭВМ.

Данный раздел посвящен анализу вредных и опасных факторов производственной среды для операторов ПЭВМ и, в частности, для метеорологов, которые будут использовать продукт в наиболее перспективном направлении его применения.

6.2. Производственная безопасность

Для обеспечения производственной безопасности необходимо проанализировать воздействия на человека вредных и опасных производственных факторов, которые могут возникать при разработке или эксплуатации проекта.

Производственный фактор считается вредным, если воздействие этого фактора на работника может привести к его заболеванию. Производственный фактор считается опасным, если его воздействие на работника может привести к его травме [35].

Все производственные факторы классифицируются по группам элементов: физические, химические, биологические и психофизические. Для данной работы целесообразно рассмотреть физические и психофизические вредные и опасные факторы производства, характерные как для рабочей зоны

программиста, как разработчика рассматриваемой в данной работе системы, так и для рабочей зоны пользователя готового продукта – метеоролога-оператора ПЭВМ. Выявленные факторы представлены в таблице 12.

Таблица 12 – Вредные и опасные производственные факторы при выполнении работ за ПЭВМ [36]

Источник фактора, наименование видов работ	Факторы (по ГОСТ 12.0.003-74)		Нормативные документы
	Вредные	Опасные	
1) Работа за ПЭВМ	1) Повышенная или пониженная температура воздуха рабочей зоны; 2) Повышенный уровень электромагнитных излучений 3) Недостаточная освещенность рабочей зоны. 4) Монотонный режим работы	1) Опасность поражения электрическим током; 2) Опасность возникновения пожара.	1) СанПиН 2.2.4.548-96; 2) СанПиН 2.2.2/2.4.1340-03; 3) СП 52.13330.2011; 4) ГОСТ Р 12.1.019-2009 ССБТ; 5) СНиП 21-01-97.

6.2.1. Вредные производственные факторы

6.2.1.1. Повышенная или пониженная температура воздуха рабочей среды

Данный фактор является вредным производственным фактором и является фактором микроклимата рабочей среды, параметры которого регулируются СанПиН 2.2.4.548-96. Он больше характерен для рабочей среды программиста-разработчика системы. К параметрам, характеризующим микроклимат в производственных помещениях, относятся:

- Температура воздуха (t , °C);
- Температура поверхностей (t , °C);
- Относительная влажность воздуха (ϕ , %);
- Скорость движения воздуха (v , м/с);

- Интенсивность теплового облучения (I , Вт/м²).

В производственных помещениях для работы с ПЭВМ происходит постоянное выделение тепла самой вычислительной техникой, вспомогательными приборами и средствами освещения. Поскольку оператор расположен в непосредственной близости с источниками выделения тепла, то данный фактор является одним из важнейших вредных факторов производственной среды оператора ПЭВМ, а высокая температура воздуха способствует быстрому перегреву организма и быстрой утомляемости [37].

Влажность оказывает большое влияние на терморегуляцию организма. Так, например, высокие показатели относительной влажности (более 85 %) затрудняют терморегуляцию снижая возможность испарения пота, низкие показатели влажности (менее 20 %) вызывают пересыхание слизистых оболочек человека [38].

Санитарные нормы устанавливают оптимальные и допустимые значения величин показателей микроклимата рабочих мест для различных категорий работ в теплый и холодный периоды года. Для программиста или оператора ПЭВМ категория работ является лёгкой (1а), т.к. работа проводится сидя, без систематических физических нагрузок. Оптимальные параметры микроклимата в офисных помещениях приведены в таблице 112.

Таблица 12 – Оптимальные параметры микроклимата производственных помещений оператора ПЭВМ

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	22–24	21–25	60–40	0,1
Теплый	23–25	22–26	60–40	0,1

Холодный период года – среднесуточная температура воздуха 10 °С и ниже, теплый период года – среднесуточная температура воздуха выше 10 °С.

В таблице 113 приведены допустимые показатели микроклимата для офисных помещений.

Таблица 13 – Допустимые показатели микроклимата производственных помещений оператора ПЭВМ [39]

Период года	Температура воздуха, °С		Температура поверхностей, °С	Относительная влажность, % воздуха	Скорость движения воздуха, м/с, для диапазона температур воздуха	
	ниже оптимальных величин	выше оптимальных величин			ниже оптимальных величин, не более	выше оптимальных величин, не более
Холодный	20,0–21,9	24,1–25,0	19–26	15–75	0,1	0,1
Теплый	21,0–22,9	25,1–28,0	20–29	15–75	0,1	0,2

6.2.1.2. Повышенный уровень электромагнитных излучений

Уровень электромагнитных излучений на рабочем месте оператора ПЭВМ является вредным фактором производственной среды, величины параметров которого определяются СанПиН 2.2.2/2.4.1340-03. Основными источниками электромагнитных излучений в помещениях для работы операторов ПЭВМ являются дисплеи компьютеров и мобильных устройств, сеть электропроводки, системный блок, устройства бесперебойного питания, блоки питания.

Излучения, применительно к дисплеям современных ПЭВМ, можно разделить на следующие классы:

- Переменные электрические поля (5 Гц – 400 кГц);
- Переменные магнитные поля (5 Гц – 400 кГц).

Воздействие данных излучений на организм человека носит необратимый характер и зависит от напряженности полей, потока энергии, частоты колебаний, размера облучаемого тела. При воздействии полей, имеющих напряженность выше предельно допустимого уровня, развиваются нарушения нервной системы, кровеносной сердечно-сосудистой системы, органов пищеварения и половой системы [40].

В таблице 14 приведены допустимые уровни параметров электромагнитных полей

Таблица 14 – Временные допустимые уровни электромагнитных полей, создаваемых ПЭВМ на рабочих местах [41]

Наименование параметров		Допустимые значения
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

6.2.1.3. Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, возникающим при работе с ПЭВМ, уровни которого регламентируются СП 52.13330.2011.

Причиной недостаточной освещенности являются недостаточность естественного освещения, недостаточность искусственного освещения, пониженная контрастность.

Работа с компьютером подразумевает постоянный зрительный контакт с дисплеем ПЭВМ и занимает от 80 % рабочего времени. Недостаточность освещения снижает производительность труда, увеличивает утомляемость и количество допускаемых ошибок, а также может привести к появлению профессиональных болезней зрения.

Разряд зрительных работ программиста и оператора ПЭВМ относится к разряду III и подразряду г (работы высокой точности). В таблице 15 представлены нормативные показатели искусственного освещения при работах заданной точности.

Таблица 15 – Требования к освещению помещений промышленных предприятий для операторов ПЭВМ [42]

Характеристика	Наименьший или	Разряд	Подразряд	Контраст	Характеристика	Искусственное освещение
----------------	----------------	--------	-----------	----------	----------------	-------------------------

зрительной работы	эквивалентный размер объекта различения, мм	зрительной работы	зрительной работы	объекта с фоном	фона	Освещённость, лк		
						При системе комбинированного освещения		При системе общего освещения
						всего	В том числе от общего	
Высокой точности	От 0,3 до 0,5	III	г	Средний, бо́льшой	Светлый, средний	400	200	200

6.2.1.4. Монотонный режим работы

При работе с ПЭВМ основным фактором, влияющим на нервную систему программиста или пользователя является огромное количество информации, которое он должен воспринимать. Это является сложной задачей, которая очень сильно влияет на сознание и психофизическое состояние из-за монотонности работы. Поэтому меры, позволяющие снизить воздействие этого вредного производственного фактора, которые регулируются СанПиН 2.2.2/2.4.1340-03, являются важными в работе оператора ПЭВМ. Они позволяют увеличить производительность труда и предотвратить появление профессиональных болезней.

Организация работы с ПЭВМ осуществляется в зависимости от вида и категории трудовой деятельности. Виды трудовой деятельности разделяются на 3 группы: группа А – работа по считыванию информации с экрана с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ПЭВМ. Работа программиста-разработчика рассматриваемой в данной работе системы относится к группам А и Б, в то время, как деятельность врача-специалиста, который будет использовать систему в профессиональной деятельности, относится к группе В. Категории трудовой деятельности различаются по степени тяжести выполняемых работ. Для снижения воздействия рассматриваемого вредного

фактора предусмотрены регламентированные перерывы для каждой группы работ – таблица 16.

Таблица 16 – Суммарное время регламентированных перерывов в зависимости от продолжительности работы, вида категории трудовой деятельности с ПЭВМ [41]

Категория работ с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное время регламентированных перерывов, мин.	
	группа А, количество знаков	группа Б, количество знаков	группа В, ч	при 8-часовой смене	при 12-часовой смене
I	до 20 000	до 15 000	до 2	50	80
II	до 40 000	до 30 000	до 4	70	110
III	до 60 000	до 40 000	до 6	90	140

6.2.2. Опасные производственные факторы

6.2.2.1. Опасность поражения электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться много внимания. Нормы электробезопасности на рабочем месте регламентируются СанПиН 2.2.2/2.4.1340-03, вопросы требований к защите от поражения электрическим током освещены в ГОСТ Р 12.1.019-2009 ССБТ.

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Опасность поражения электрическим током усугубляется тем, что человек не в состоянии без специальных приборов обнаружить напряжение дистанционно.

Помещение, где расположено рабочее место оператора ПЭВМ, относится к помещениям без повышенной опасности ввиду отсутствия следующих факторов: сырость, токопроводящая пыль, токопроводящие полы,

высокая температура, возможность одновременного прикосновения человека к имеющим соединение с землей металлоконструкциям зданий, технологическим аппаратам, механизмам и металлическим корпусам электрооборудования.

Для оператора ПЭВМ при работе с электрическим оборудованием обязательны следующие меры предосторожности:

- Перед началом работы нужно убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей;
- При обнаружении неисправности оборудования и приборов необходимо, не делая никаких самостоятельных исправлений, сообщить человеку, ответственному за оборудование [41, 43].

6.2.2.2. Опасность возникновения пожара

Возникновение пожара является опасным производственным фактором, т.к. пожар на предприятии наносит большой материальный ущерб, а также часто сопровождается травмами и несчастными случаями. Регулирование пожаробезопасности производится СНиП 21-01-97.

В помещениях с ПЭВМ повышен риск возникновения пожара из-за присутствия множества факторов: наличие большого количества электронных схем, устройств электропитания, устройств кондиционирования воздуха; возможные неисправности электрооборудования, освещения, или неправильная их эксплуатация может послужить причиной пожара.

Возможные виды источников воспламенения:

- Искра при разряде статического электричества;
- Искры от электрооборудования;
- Искры от удара и трения;
- Открытое пламя [44].

6.2.3. Мероприятия и рекомендации по устранению и минимизации

Для поддержания нормальных значений параметров микроклимата на рабочих местах рекомендуется оснащать их системами отопления, вентиляции и кондиционирования воздуха. Также, в некоторых случаях, целесообразно обеспечить питьевое водоснабжение. В помещениях для работы с ПЭВМ

должна производиться ежедневная влажная уборка, а также систематическое проветривание после каждого часа работы [39].

Для защиты операторов ПЭВМ от негативного воздействия электромагнитных полей в первую очередь необходимо, чтобы используемая техника удовлетворяла нормам и правилам сертификации. При работе с ПЭВМ установлены регламентированные перерывы, а также иногда предусмотрено использование экранов и фильтров в целях защиты оператора [41].

Для создания и поддержания благоприятных условий освещения для операторов ПЭВМ, их рабочие места должны соответствовать санитарно-эпидемиологическим правилам СанПиН 2.2.2/2.4.1340-03. Рабочее помещение должно иметь естественное и искусственное освещение, соответствующее показателям, представленным в таблице **Ошибка! Источник ссылки не найден..** Для рассеивания естественного освещения следует использовать жалюзи на окнах рабочих помещений. В качестве источников искусственного освещения должны быть использованы люминесцентные лампы, лампы накаливания – для местного освещения [42].

Для предупреждения преждевременной утомляемости пользователей ПЭВМ рекомендуется организовывать рабочую смену путем чередования работ с использованием ПЭВМ и без него. В случаях, когда характер работы требует постоянного взаимодействия с компьютером (работа программиста-разработчика) с напряжением внимания и сосредоточенности, при исключении возможности периодического переключения на другие виды трудовой деятельности, не связанные с ПЭВМ, рекомендуется организация перерывов на 10–15 мин. через каждые 45–60 мин. работы. При высоком уровне напряженности работы рекомендуется психологическая разгрузка в специально оборудованных помещениях [41].

К мероприятиям по предотвращению возможности поражения электрическим током относятся:

- При производстве монтажных работ необходимо использовать только исправный инструмент, аттестованный службой КИПиА;

- С целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены;
- При включенном сетевом напряжении работы на задней панели должны быть запрещены;
- Все работы по устранению неисправностей должен производить квалифицированный персонал;
- Необходимо постоянно следить за исправностью электропроводки [41, 43].

Для профилактики организации действий при пожаре должен проводиться следующий комплекс организационных мер: должны обеспечиваться регулярные проверки пожарной сигнализации, первичных средств пожаротушения; должен проводиться инструктаж и тренировки по действиям в случае пожара; не должны загромождаться или блокироваться пожарные выходы; должны выполняться правила техники безопасности и технической эксплуатации электроустановок; во всех служебных помещениях должны быть установлены «Планы эвакуации людей при пожаре и других ЧС», регламентирующие действия персонала при возникновении пожара.

Для предотвращения пожара помещение с ПЭВМ должно быть оборудовано первичными средствами пожаротушения: углекислотными огнетушителями типа ОУ-2 или ОУ-5; пожарной сигнализацией, а также, в некоторых случаях, автоматической установкой объемного газового пожаротушения [44].

6.3. Экологическая безопасность

В данном разделе рассматривается воздействие на окружающую среду деятельности по разработке проекта, а также самого продукта в результате его реализации на производстве.

Разработка программного обеспечения и работа за ПЭВМ не являются экологически опасными работами, потому объект, на котором производилась

разработка продукта, а также объекты, на которых будет производиться его использование операторами ПЭВМ относятся к предприятиям пятого класса, размер селитебной зоны для которых равен 50 м [45].

Программный продукт, разработанный непосредственно в ходе выполнения магистерской диссертации, не наносит вреда окружающей среде. Но в связи с тем, что в ходе операций над входными данными возможны погрешности вычисления, ошибки реализации вычислений программистом, а также ошибки во входных данных, работа с программным продуктом может привести к неправильным выводам метеоролога. Поэтому требуется тщательная проверка адекватности всех вычислений программного продукта метеорологом.

Средства, необходимые для разработки и эксплуатации программного комплекса могут наносить вред окружающей среде.

Современные ПЭВМ производят практически без использования вредных веществ, опасных для человека и окружающей среды. Исключением являются аккумуляторные батареи компьютеров и мобильных устройств. В аккумуляторах содержатся тяжелые металлы, кислоты и щелочи, которые могут наносить ущерб окружающей среде, попадая в гидросферу и литосферу, если они были неправильно утилизированы. Для утилизации аккумуляторов необходимо обращаться в специальные организации, специализировано занимающиеся приемом, утилизацией и переработкой аккумуляторных батарей [46].

Люминесцентные лампы, применяющиеся для искусственного освещения рабочих мест, также требуют особой утилизации, т.к. в них присутствует от 10 до 70 мг ртути, которая относится к чрезвычайно-опасным химическим веществам и может стать причиной отравления живых существ, а также загрязнения атмосферы, гидросферы и литосферы. Сроки службы таких ламп составляют около 5-ти лет, после чего их необходимо сдавать на переработку в специальных пунктах приема. Юридические лица обязаны

сдавать лампы на переработку и вести паспорт для данного вида отходов [46-48].

6.4. Безопасность в чрезвычайных ситуациях

В рабочей среде оператора ПЭВМ возможно возникновение следующих чрезвычайных ситуаций техногенного характера:

- Пожары и взрывы в зданиях и на коммуникациях;
- Внезапное обрушение зданий.

Среди возможных стихийных бедствий можно выделить метеорологические (ураганы, ливни, заморозки), гидрологические (наводнения, паводки, подтопления), природные пожары.

К чрезвычайным ситуациям биолого-социального характера можно отнести эпидемии, эпизоотии, эпифитотии.

Экологические чрезвычайные ситуации могут быть вызваны изменениями состояния, литосферы, гидросферы, атмосферы и биосферы в результате деятельности человека [38].

Наиболее характерной для объекта, где размещаются рабочие помещения, оборудованные ПЭВМ, чрезвычайной ситуацией является пожар.

Помещение для работы операторов ПЭВМ по системе классификации категорий помещений по взрывопожарной и пожарной опасности относится к категории Д (из 5-ти категорий А, Б, В1-В4, Г, Д), т.к. относится к помещениям с негорючими веществами и материалами в холодном состоянии [49].

Каждый сотрудник организации должен быть ознакомлен с инструкцией по пожарной безопасности, пройти инструктаж по технике безопасности и строго соблюдать его.

Запрещается использовать электроприборы в условиях, не соответствующих требованиям инструкций изготовителей, или имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару, а также эксплуатировать электропровода и кабели с поврежденной или потерявшей защитные свойства изоляцией. Электроустановки и бытовые электроприборы в помещениях по окончании

рабочего времени должны быть обесточены (вилки должны быть вынуты из розеток). Под напряжением должны оставаться дежурное освещение и пожарная сигнализация. Недопустимо хранение легковоспламеняющихся, горючих и взрывчатых веществ, использование открытого огня в помещениях офиса.

Перед уходом из служебного помещения работник обязан провести его осмотр, закрыть окна, и убедиться в том, что в помещении отсутствуют источники возможного возгорания, все электроприборы отключены и выключено освещение. С периодичностью не реже одного раза в три года необходимо проводить замеры сопротивления изоляции токоведущих частей силового и осветительного оборудования.

Повышение устойчивости достигается за счет проведения соответствующих организационно-технических мероприятий, подготовки персонала к работе в ЧС [38].

Работник при обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры и т.п.) должен:

- Немедленно прекратить работу и вызвать пожарную охрану по телефону «01», сообщив при этом адрес, место возникновения пожара и свою фамилию;
- Принять по возможности меры по эвакуации людей и материальных ценностей;
- Отключить от сети закрепленное за ним электрооборудование;
- Приступить к тушению пожара имеющимися средствами пожаротушения;
- Сообщить непосредственному или вышестоящему начальнику и оповестить окружающих сотрудников;
- При общем сигнале опасности покинуть здание согласно «Плану эвакуации людей при пожаре и других ЧС».

Для тушения пожара применять ручные углекислотные огнетушители (типа ОУ-2, ОУ-5), находящиеся в помещениях офиса, и пожарный кран

внутреннего противопожарного водопровода. Они предназначены для тушения начальных возгораний различных веществ и материалов, за исключением веществ, горение которых происходит без доступа воздуха. Огнетушители должны постоянно содержаться в исправном состоянии и быть готовыми к действию. Категорически запрещается тушить возгорания в помещениях офиса при помощи химических пенных огнетушителей (типа ОХП-10) [50].

6.5. Правовые и организационные вопросы обеспечения безопасности

6.5.1. Правовые нормы трудового законодательства для рабочей зоны оператора ПЭВМ

Регулирование отношений между работником и работодателем, касающихся оплаты труда, трудового распорядка, особенности регулирования труда женщин, детей, людей с ограниченными способностями и проч., осуществляется законодательством РФ, а именно трудовым кодексом РФ.

Продолжительность рабочего дня не должна быть меньше указанного времени в договоре, но не больше 40 часов в неделю. Для работников до 16 лет – не более 24 часов в неделю, от 16 до 18 лет и инвалидов I и II группы – не более 35 часов.

Возможно установление неполного рабочего дня для беременной женщины; одного из родителей (опекуна, попечителя), имеющего ребенка в возрасте до четырнадцати лет (ребенка-инвалида в возрасте до восемнадцати лет). Оплата труда при этом производится пропорционально отработанному времени, без ограничений оплачиваемого отпуска, исчисления трудового стажа и других прав.

При работе в ночное время продолжительность рабочей смены сокращается на один час. К работе в ночную смену не допускаются беременные женщины; работники, не достигшие возраста 18 лет; женщины, имеющие детей в возрасте до трех лет, инвалиды, работники, имеющие детей-инвалидов, а также работники, осуществляющие уход за больными членами их семей в соответствии с медицинским заключением, матери и отцы-одиночки детей до пяти лет.

Организация обязана предоставлять ежегодный отпуск продолжительностью 28 календарных дней. Дополнительные отпуска предоставляются работникам, занятым на работах с вредными или опасными условиями труда, работникам имеющими особый характер работы, работникам с ненормированным рабочим днем и работающим в условиях Крайнего Севера и приравненных к нему местностях.

В течение рабочего дня работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Всем работникам предоставляются выходные дни, работа в выходные дни осуществляется только с письменного согласия работника.

Организация-работодатель выплачивает заработную плату работникам. Возможно удержание заработной платы только в случаях установленных ТК РФ ст. 137. В случае задержки заработной платы более чем на 15 дней, работник имеет право приостановить работу, письменно уведомив работодателя.

Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд [51].

6.5.2. Организационные мероприятия при компоновке рабочей зоны

К мероприятиям, относящимся к компоновке рабочей зоны относятся работы по организации рабочего места пользователя, позволяющие наилучшим образом организовать деятельность работника, делая его работу максимально удобной и безопасной.

Основным направлением использования разработанной программной системы является формирование скрытых групп на основе анализа среднемесячных температур за большой период, что позволяет изучать глобальные изменения климата во времени, а также прогнозировать и изучать текущие его изменения. Сам продукт не влияет на организацию рабочей зоны, однако работа с ним позволит реорганизовать работу специалистов, что в свою

очередь повлияет на организацию рабочей зоны. Это может быть охарактеризовано с помощью следующих факторов:

- Повышение производительности математических вычислений метеорологических данных, что позволит метеорологу быстрее получать запрашиваемые данные;
- Сокращение умственных и зрительных нагрузок метеоролога за счет автоматизации анализа вычислений данных;
- Работа программного продукта позволяет использовать вычислительные мощности других компьютеров, объединяя в вычислительную сеть подключенные компьютеры. Таким образом все доступные для специалиста компьютеры, а также удаленно подключенные в сеть могут быть использованы для вычислений и ускорения получения данных.

Все перечисленные факторы повышают, облегчают работу и положительно сказываются на производительности труда.

Заключение

В результате выполнения выпускной квалификационной работы был проведен сравнительный обзор и анализ методов структуризации и построения распределенных вычислительных систем. Основное внимание было уделено методам кластерного анализа и практике построения распределенных грид систем.

В ходе исследования были проведены программные эксперименты по оценке эффективности выполнения вычислений в задаче фазовой кластеризации. Рассматривались следующие модели выполнения вычислений:

- однопоточное вычисление в стандартном режиме;
- многопоточное вычисление в стандартном режиме;
- многопоточное вычисление на базе суперкомпьютера ТПУ;
- мультиагентные вычисления на базе распределенной вычислительной системы.

Анализ эффективности показал, что алгоритм фазовой группировки с использованием метеорологических данных поддается распараллеливанию, и в многопоточной реализации обеспечивает увеличение производительности на 25-30%.

Дополнительно, была спроектирована и реализована многомашинная распределенная вычислительная среда из персональных компьютеров для формирования скрытых функционально-детерминированных структур для классификации и анализа геофизических данных.

Технологическая схема реализации и развертывания многомашинной системы включала в себя два функциональных компонента: центр управления и масштабируемый кластер вычислительных агентов, предоставляющих свои вычислительные мощности для выполнения задач центра по запросу. Основным каналом связи между компонентами системы являлся сокетный интерфейс.

Кроме того, в выпускной квалификационной работе рассмотрены вопросы финансового менеджмента, ресурсоэффективности и ресурсосбережения, а также идентифицированы основные опасные и вредные производственные факторы, мероприятия по их устранению, и правовые и организационные вопросы обеспечения безопасности.

Список публикаций студента

1. Демин А. Ю. , Сорокин В. А. Мультиагентное моделирование подводной обстановки и её трехмерная визуализация // Фундаментальные исследования. - 2014 - №. 12-3. - С. 483-486
2. Сорокин В. А. , Демин А. Ю. Разработка программного комплекса для визуализации 3D модели рельефа поверхности // Перспективы развития информационных технологий : труды Всероссийской молодежной научно-практической конференции, Кемерово, 29-30 Мая 2014. - Кемерово: ГОУ КузГТУ, 2014 - С. 353-354
3. Сорокин В.А., Рохмистров Д.С., Ботыгин И.А. Разработка и исследование итеративного параллельного алгоритма формирования скрытых функционально-детерминированных структур для классификации и анализа метеорологических данных // Технологии Microsoft в теории и практике программирования: Сборник трудов XIII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 22-23 марта 2016. – Томск: Изд-во ТПУ.
4. Проектирование 3D модель рельефа поверхности с помощью WPF В.А.Сорокин, А.Ю.Демин // Технологии Microsoft в теории и практике программирования: Сборник трудов XI Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 23-24 апреля 2014. – Томск: Изд-во ТПУ.
5. Программный комплекс для трехмерной визуализации подводно обстановки и моделирования поведения группы АНПА В.А. Сорокин, А.Ю. Демин // Молодежь и современные информационные технологии: Сборник трудов XII Международной научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 12-14 ноября 2014. – Томск: Изд-во ТПУ.
6. Численная реализация алгоритма фазовой группировки на примере температурных рядов В.А. Сорокин, Д.С. Рохмистров, Ю.В. Волков, П.Б. Завьялов // Молодежь и современные информационные технологии:

Сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 9-13 ноября 2015. – Томск: Изд-во ТПУ.

7. Ботыгин И.А., Гордов Е.П., Зензин А.С., Рохмистров Д.С., Сорокин В.А. Математическое и программное обеспечение мультиагентной универсальной децентрализованной масштабируемой информационно-вычислительной системы // Информационные технологии в науке, управлении, социальной сфере и медицине: Сборник трудов III международной научной конференции, Томск, 23-26 мая 2016. – Томск: Изд-во ТПУ.
8. Демин А.Ю., Сорокин В.А., Анферов И.А., Хамухин А.А. Визуализация процесса имитационного моделирования группы автономных необитаемых подводных аппаратов // Известия ЮФУ, Январь 2016 – Таганрог: ФГАОУ ВО ЮФУ, 2016, С. 77-87.
9. Moiseev A. N. , Demin A. Y. , Dorofeev V. A. , Sorokin V. A. Discrete-event approach to simulation of queueing networks // Key Engineering Materials . - 2016 - Vol. 685. - p. 939-942.
10. Sorokin V. A. , Demin A. Y. , Khasaeva D. Z. Three-dimensional visualization of the underwater environment using graphical library "3Dbodies" // Mechanical Engineering, Automation and Control Systems: Proceedings of International Conference, Tomsk, October 16-18, 2014. - Tomsk: TPU Publishing House, 2014 - p. 1-5.

Список использованных источников

1. Задачи для суперкомпьютеров, суперкомпьютерные приложения // Parallel.ru URL: <https://parallel.ru/research/challenges.html> (дата обращения: 02.06.2016).
2. Параллельные вычисления при решении современных задач науки и техники // Вычислительная механика. Кафедра механико-математического факультета МГУ им М.В. Ломоносова URL: http://compmech.math.msu.su/publ/Zabrodin_Parallel_comp.pdf (дата обращения: 02.06.2016).
3. Учебное пособие. Параллельное программирование на основе библиотек. / Васильева М.В., Захаров П.Е., Сирдитов И.К., Попов П.А., Еремеева М.С., Якутск: Издательско-полиграфический комплекс СВФУ, 2011.
4. Суперкомпьютер // Wikipedia URL: <https://ru.wikipedia.org/wiki/Суперкомпьютер> (дата обращения: 02.06.2016).
5. Кластер (группа компьютеров) // Wikipedia URL: [https://ru.wikipedia.org/wiki/Кластер_\(группа_компьютеров\)](https://ru.wikipedia.org/wiki/Кластер_(группа_компьютеров)) (дата обращения: 02.06.2016).
6. Грид // Wikipedia URL: <https://ru.wikipedia.org/wiki/Грид> (дата обращения: 02.06.2016).
7. Волкова С.Л., Воробьева И.В. Исследование алгоритмов кластеризации // V Международная студенческая электронная научная конференция «Студенческий научный форум»: материалы. М.: Российская академия естествознания, 2013.
8. Нейский И.М. Классификация и сравнение методов кластеризации. // Научно-образовательный кластер CLAIM URL: http://it-claim.ru/Persons/Neyskiy/Article2_Neiskiy.pdf (дата обращения: 03.06.2016).
9. K-means // Wikipedia URL: <https://ru.wikipedia.org/wiki/K-means> (дата обращения: 03.06.2016).

- 10.Кластеризация категориальных данных: масштабируемый алгоритм CLOPE
// BaseGroups Labs: Технологии анализа данных URL:
<https://basegroup.ru/community/articles/clope> (дата обращения: 03.06.2016).
- 11.Data Mining // ИНТУИТ URL:
<http://www.intuit.ru/studies/courses/6/6/lecture/184?page=2> (дата обращения:
03.06.2016).
- 12.Речкалов Т.В. Параллельный алгоритм кластеризации для многоядерного
сопроцессора Intel Xeon Phi // Суперкомпьютерные дни в России 2015:
материалы. М.: Издательство МГУ, 2015.
- 13.Самоорганизующиеся карты Кохонена — математический аппарат //
BaseGroup Labs Технологии анализ данных URL:
<https://basegroup.ru/community/articles/som> (дата обращения: 03.06.2016).
- 14.Селезнев К. Проблемы анализа Больших Данных // Открытые системы.
2012. №7.
- 15.Распределенные системы. Принципы и парадигмы. / Таненбаум Э., Спб.:
Питер, 2003.
- 16.Введение в распределенные вычисления. Учебное пособие. / Косяков
М.С., Спб.: НИУ ИТМО, 2014.
- 17.Распределенные вычислительные системы. / Радченко Г.И., Челябинск:
Фотохудожник, 2012.
- 18.Введение в грид технологии / Радченко Г.И., Демичев А.П., Ильин В.А.,
Крюков А.П. и др. М.: НИИЯФ МГУ, 2007.
- 19.Великевич С., Демченко Ю. Использование Грид-технологий для
построения распределенных САПР // RELARN-2005: Материалы. М.:
Ассоциация РЕЛАРН, 2005.
- 20.Голубовский А.В., Гиоргизова-Гай В.Ш. Построение одноуровневой
Грид-системы на платформе Condor. // Системный анализ и
информационные технологии: материалы. Киев: 2011.

21. Воеводин В.В., Жолудев Ю.А., Соболев С.И., Стефанов К.С. Эволюция системы метакомпьютинга X-COM // Вестник Нижегородского университета им. Н.И. Лобачевского. 2009. №4.
22. Березовский П.С. Реализация системы диспетчеризации заданий SARD в одноуровневом гриде // Препринты ИПМ им. М.В.Келдыша. 2010. № 49. 32 с. URL: <http://library.keldysh.ru/preprint.asp?id=2010-49>
23. Методические указания. Установка и настройка компонентов вычислительной среды OurGrid / Бороздюхин А.А., Саратов: Саратовский государственный технический университет, 2007.
24. Увеличение вычислительной мощности распределенных систем с помощью грид-систем из персональных компьютеров // Сайт Российского отделения международной федерации грид-систем из персональных компьютеров URL: http://desktopgrid.ru/articles/increasing_computing_power.pdf (дата обращения: 03.06.2016).
25. Кластеризация // MachineLearning.ru URL: <http://www.machinelearning.ru/wiki/index.php?title=Кластеризация> (дата обращения: 26.02.2016).
26. Обзор алгоритмов кластеризации данных // Хабрахабр URL: <https://habrahabr.ru/post/101338/> (дата обращения: 26.02.2016).
27. Ю.В. Волков, В.А. Тартаковский Региональная группировка данных, полученных на метеостанциях Евразии // Энергетика: Эффективность, надежность, безопасность. - Томск: ООО "СКАН", 2013. – С. 177-178.
28. Вакман Д.Е., Вайнштейн Л.А. Амплитуда, фазы частота - основные понятия теории колебаний // Успехи физических наук. 1977. 123. Вып.4. С. 657.
29. В. О. Попова, Ю. В. Волков; науч. рук. Ю. В. Волков // Неразрушающий контроль: электронное приборостроение, технологии, безопасность: сборник трудов IV Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных, Томск, 26-30 мая 2014 г. в 2 т. /

- Национальный исследовательский Томский политехнический университет (ТПУ). — 2014. — Т. 2. — [С. 211-213].
30. В.А. Тартаковский, В.А. Крутиков, Ю.В. Волков, Н.Н. Чередыко
Классификация климата путем анализа фазы температурных рядов // Оптика атмосферы и океана. - 2015. - №8. - С. 711-718.
31. Ю.В. Волков, В.А. Тартаковский Региональная группировка данных, полученных на метеостанциях Евразии // Энергетика: Эффективность, надежность, безопасность. - Томск: ООО "СКАН", 2013. - С. 177-178.
32. Вакман Д.Е., Вайнштейн Л.А. Амплитуда, фазы частота - основные понятия теории колебаний // Успехи физических наук. 1977. 123. Вып.4. С.657.
33. Vakman D. On the analytic signal, the Teager-Kaiser energy algorithm, and other methods for defining amplitude and frequency // IEEE Trans. Signal processing. 1996. 44. № 4. P.791.
34. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. М: Мир, 1978.
35. Охрана труда. Основы безопасности жизнедеятельности // www.Grandars.ru. 2016. URL: <http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/ohrana-truda.html> (дата обращения: 22.04.2016).
36. ГОСТ 12.0.003-74. Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация // Библиотека ГОСТов. 2016. URL: <http://vsegost.com/Catalog/41/41131.shtml> (дата обращения: 22.04.2016).
37. Ефремова О. С. Требования охраны труда при работе на персональных электронно-вычислительных машинах. — 2-е изд., перераб. и доп. — М. : Издательство «Альфа-Пресс», 2008. — 176 с.
38. Назаренко О. Б. Безопасность жизнедеятельности: учебное пособие / О. Б. Назаренко, Ю. А. Амелькович; Томский политехнический университет. — 3-е изд., перераб. и доп. — Томск: Изд-во Томского политехнического университета, 2013. — 178 с.

- 39.СанПиН 2.2.4.548-96. Санитарные правила и нормы. Гигиенические требования к микроклимату производственных помещений // Библиотека гостов и нормативов. 2016. URL: http://ohranatruda.ru/ot_biblio/normativ/data_normativ/5/5225/ (дата обращения: 23.04.2016).
- 40.Белов С. В. Безопасность жизнедеятельности и защита окружающей среды (техносферная безопасность): учебник / С. В. Белов. – 2-е изд., испр. и доп. – М.: Издательство Юрайт, 2011. – 680 с.
- 41.СанПиН 2.2.2/2.4.1340-03. Санитарно-эпидемиологические правила и нормы. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы // Библиотека гостов и нормативов. 2016. URL: http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/39/39082/#i72870 (дата обращения: 23.04.2016).
- 42.СП 52.13330.2011. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95 // Докипедия. 2016. URL: <http://dikipedia.ru/document/5147250> (дата обращения: 23.04.2016).
- 43.ГОСТ Р 12.1.019-2009 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты // Электронный фонд правовой и нормативно-технической документации. 2010. URL: <http://docs.cntd.ru/document/gost-r-12-1-019-2009-ssbt> (дата обращения: 24.04.2016).
- 44.СНиП 21-01-97. Пожарная безопасность зданий и сооружений // Библиотека гостов и нормативов. 2016. URL: http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/2/2107/ (дата обращения: 24.04.2016).
- 45.СанПиН 2.2.1/2.1.1.1200-03. Санитарно-эпидемиологические правила и нормативы. Санитарно-защитные зоны и санитарная классификация предприятий, сооружений и других объектов // Библиотека гостов и нормативов. 2016. URL:

- http://ohranatruda.ru/ot_biblio/normativ/data_normativ/11/11774/ (дата обращения: 24.04.2016).
46. СанПиН 2.1.7.1322-03. Санитарно-эпидемиологические правила и нормативы. Гигиенические требования к размещению и обезвреживанию отходов производства и потребления. 2.1.7. Почва, очистка населённых мест, бытовые и промышленные отходы, санитарная охрана почвы // Библиотека гостей и нормативов. 2016. URL: http://ohranatruda.ru/ot_biblio/normativ/data_normativ/11/11774/ (дата обращения: 02.05.2016).
47. Постановление Правительства РФ от 03.09.2010 N 681 (ред. от 01.10.2013) "Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде // Консультант Плюс. 2015. URL: http://www.consultant.ru/document/cons_doc_LAW_104420/e1b31c36ed1083efeb6cd9c63ed12f99e2ca77ed/#dst100007 (дата обращения: 02.05.2016).
48. Энергосбережение в компьютерном мире // НWP. 2008. URL: http://www.hwp.ru/articles/Energoberezhenie_v_kompyuternom_mire_CHast_1___osnovnie_tendentsii/?SHOWALL_1=1 (дата обращения: 24.04.2016).
49. НПБ 105-03 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности // Электронный фонд правовой и нормативно-технической документации. 2016. URL: <http://docs.cntd.ru/document/1200032102> (дата обращения: 24.04.2016).
50. ППБ 01–03. Правила пожарной безопасности в Российской Федерации. – М.: Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий, 2003.

- 51.Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ (ред. от 30.12.2015) // Консультант Плюс. 2015. URL: http://www.consultant.ru/document/cons_doc_law_34683/?utm_campaign=law_doc&utm_source=google.adwords&utm_medium=cpc&utm_content=Labor%20Code&gclid=CjwKEAjwtgPe4BRCB66GG8PO69QkSJAC4EhHhU-5yAFZCJfmzkTLNGnrpgHhAYFPhhPzRo-sZGWmqnBoCPynw_wcB (дата обращения: 25.04.2016).
- 52.В.Ю. Конотопский Методические указания к выполнению раздела «Финансовый менеджмент, Ресурсоэффективность и ресурсосбережение» магистерских диссертаций для всех специальностей ИК. - Томск: Издательство Томского политехнического университета, 2015.
- 53.Кнышова Е. Н. Экономика организации: учебник / Е. Н. Кнышова, ЕЕ. Панфилова. – Москва: Форум Инфра-М, 2012. – 334 с.: ил. – Профессиональное образование.
- 54.Бочаров В. В. Инвестиции: учебник для вузов / В. В. Бочаров. – 2-е изд. – СПб.: Питер, 2009. – 381 с. – Учебник для вузов.
- 55.Староверова Г. С. Экономическая оценка инвестиций : учебное пособие / Г. С. Староверова, А. Ю. Медведев, И. В. Сорокина. – 2-е изд., стер. – Москва: КноРус, 2009. — 312 с
- 56.Несветаев Ю. А. Экономическая оценка инвестиций: учебное пособие / Ю. А. Несветаев; Московский Государственный индустриальный университет; Институт дистанционного образования. – 3-е изд., стер. – Москва: Изд-во МГИУ, 2006. – 162 с.
- 57.Шульмин В. А. Экономическое обоснование в дипломных проектах: учебное пособие для вузов / В. А. Шульмин, Т. С. Усынина. – Старый Оскол: ТНТ, 2012. – 192 с.
- 58.Голосовский С. И. Эффективность научных исследований в промышленности / С. И. Голосовский. – Москва: Экономика, 1986. – 159 с.

59. Мигуренко Р. А. Научно-исследовательская работа: учебно-методическое пособие / Р. А. Мигуренко; Национальный исследовательский Томский политехнический университет (ТПУ), Институт дистанционного образования (ИДО). – 2-е изд., стер. – Томск: Изд-во ТПУ, 2010. – 184 с.

Приложение А

Раздел 1

Сравнительный обзор и анализ методов структуризации и построения распределенных вычислительных систем

Студент:

Группа	ФИО	Подпись	Дата
8BM4Б	Сорокин Василий Александрович		

Консультант кафедры ИПС:

Должность	ФИО	Ученая степень, звание	Подпись	Дата

Консультант – лингвист кафедры ИПС:

Должность	ФИО	Ученая степень, звание	Подпись	Дата

A comparative review and analysis of methods for structuring and building distributed computing systems

Methods of cluster analysis review

One of the approaches to processing a large volume of geophysical data (in particular, meteorological data) is the cluster analysis.

Clustering (or cluster analysis) is the task of partitioning a set of objects into groups called clusters.

Use of the cluster analysis in general includes the following stages:

6. Selection of sampling objects for clustering.
7. Determination of a set of variables, which will be evaluated by the objects in a sample. Normalization of variables, if necessary.
8. Calculation of the measure of similarity between objects.
9. Application of the cluster analysis to create groups of similar objects (clusters).
10. Presentation of the analysis results.

After receiving and analyzing the results, selected metric and clustering method can be adjusted in order to obtain the optimal result.

According to the method of data the processing methods of cluster analysis can be divided into hierarchical and non-hierarchical groups. Hierarchical methods perform merging of smaller clusters into large (agglomerative) or split large clusters into smaller (divisive) [7].

It is also possible to allocate some features of the methods, for example, a method of analysis: clear or fuzzy, a number of applications for clustering algorithms: with a single-stage clustering or multi-step clustering [8].

The algorithm BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) provides a two-step clustering process and is used for very large datasets. This algorithm works only with the numeric data.

The K-means method (k-means) divides the set of elements of a vector space to a certain number of clusters k . The algorithm is simple to be used, however,

sensitive to outliers, which can skew the average and its disadvantages include the necessity of knowing the number of clusters [9].

The CLOPE algorithm (Clustering with slope) is designed to work with transactional data, has a high scalability and speed, as well as the clustering quality. CLOPE automatically selects the required number of clusters and is widely used in data mining [10].

The algorithm PAM (Partitioning around medoids) is a modification of the k-means algorithm. The Algorithm is less sensitive to noise and outliers than the k-means algorithm. PAM is effective for small databases; it should not be used for large data sets [11]. It is used in a wide range of knowledge fields such as text mining, bioinformatics, intelligent transportation systems [12].

Kohonen self-organizing maps – a type of neural network algorithms that uses a method without a teacher and represents one of the variants of clustering of high-dimensional vectors. This algorithm can be considered as one of the methods of projecting the multidimensional space into a space with a lower dimensionality. This technique can be used to search and analyze patterns in the source data [13].

Thus, there is no single universal clustering algorithm. Using any algorithm, it is important to understand its strengths and weaknesses, to consider the nature of the data with which it works best and its ability to scalability.

The problem of big data clustering is that the available algorithms suggest the possibility of a direct reference to any information entity of the basic data (impossible to predict which entities the algorithm needs). The raw data may be distributed over the multiple servers, and there is no guarantee that each cluster is kept strictly on the one server. If the data is distributed across the servers it is transparent for a clustering algorithm (it considers that the data are arranged in a distributed virtual memory), it will inevitably lead to a copying a large data from one server to another.

The solution may be as follows. Each server runs its own algorithm, which operates only the server data, and provides output parameters of the founded clusters and their measured weight based on the number of elements in the cluster. Then, the resulting information is collected on a central server and made metaclustering -

allocation of groups of closely spaced clusters based on their weights. This method is versatile, well parallelized and can use any other clustering algorithms, but it requires a serious research, testing on real data, and comparing the results obtained with other "local" methods.

Thus, for the analysis of the data which is greater than the vast majority of clustering methods inapplicable in its pure form, and more research and greater computational power is needed [14].

4. Distributed grid systems overview

One of the most promising and affordable way of the organization metacomputing is to join autonomous computing resources (computers) into a single virtual computing node (grid). Distributed grid systems are a type of parallel computing which is based on the desktop computers and compares favorably with conventional supercomputers that can be deployed using conventional protocols as Ethernet. Thus, desktop computers can be combined into a single distributed system, and their resources can be used for calculations.

One of the most important characteristic of distributed systems is that differences between computers and communication means between them are hidden from the users. Another important feature is the way in which users and applications uniformly operate in distributed systems, no matter when and where there is their interaction. Distributed systems must also be relatively easy to expand or scale [15].

The main distinguishing features of distributed systems include:

- Absence of a single time for the distributed components of the system characterizes the spatial distribution of the components of the system, namely processors, included in its composition, which implies the absence of synchronicity in their work;
- Absence of shared memory. Key feature, which implies the need for the exchange of messages between software components of distributed systems for their communication and synchronization.
- Geographical distribution;

- Independence and heterogeneity. Computers in a distributed system loosely bound in the sense that they can have a different composition and different performance and therefore provide a different runtime for identical tasks. Usually they are not part of a specialized system, but operate together, providing their services to each other to perform common tasks [16].

The main objective of distributed systems – facilitate access to remote resources for the users and ensure their sharing, regulating this process [15]. Efficient distribution system should have the following properties: transparency, openness, security, scalability [16].

the **transparency** of the distributed system implies its ability to hide their distributed nature, namely, distribution of processes and resources across multiple computers, and presented to users and application developers in the form of a single, centralized computer system. In order the distributed computing system to be presented to the user as a single system, the following types of transparency is used

- Transparent access to resources - difference in the data representation and how to access the resources of the computing system should be hidden from the users;
- Transparent location of resources – physical location of required resource should be unimportant for the user;
- Replication – hiding from the user the fact that in reality there is more than one copy of the resources used;
- Parallel access – possibility of joint (simultaneous) use of the same resources by different users, regardless pyr other. At the same time the fact of sharing the resource should remain hidden from the user;
- Transparency of failure – failure (disconnection) any resources should not have an effect on the user and his application [17].

Open distributed system – a system that offers the service, which calling requires a standard syntax and semantics. In distributed service systems are usually determined through interfaces. Properly described, interface definition allows the possibility of joint work of an arbitrary process in need of such an interface with

another arbitrary process, providing the interface. Defining the interface also allows two independent groups to create completely different implementations of this interface for two different distributed systems that will work exactly the same. Also, an open system must have: *the ability to interact*, that characterizes how the implementation of two systems or components from different manufacturers are able to work together, relying only on what services each of them correspond to a common standard; *portability*, that describes how the application is designed for a distributed system A can be performed without any changes in a distributed system B, realizing the same as in A and interfaces; *flexibility* - easy system configuration, consisting of various components [15].

Scalability – the ability of a computer system to effectively deal with the increasing number of supported users or resources without compromising performance and without increasing the administrative load on its management. The system is called scalable if it is able to increase its productivity by adding new hardware.

The scalability of distributed systems is determined in the following areas:

- Load scalability - the system's ability to increase its productivity when increasing the load by replacing existing hardware components (vertical scale) with more powerful or by adding new hardware (horizontal scaling);
- Geographic scalability - the system's ability to maintain its main characteristics, such as performance, simplicity and ease of use, with the territorial distribution of its components from a local mutual arrangement to a more distributed;
- Administrative scalability. It characterizes the ease of system management by increasing the number of independent administrative institutions serving part of a distributed system.

The concept of the grid does not appear as an abstract idea, but as a response to the emerging needs of large information and computing resources which can be dynamically allocated to solve the tasks in scientific, industrial, administrative and

commercial areas. Creating a grid environment involves the distribution of computational resources on geographically separated nodes that have installed the special software to distribute tasks over the nodes and take them there, return the results to the user, to control the rights of users to have an access to certain resources, monitoring resources and so on. Public resources may include:

- **Computing resources** provide the user's task of grid-system processing power. Computing resources can be clusters and individual workstations. The main characteristic of the computing resource is a performance;
- **Storage resources** are use application software that implements unified interface for management and data transfer. The main characteristic of storage resources is their volume;
- **Information resources** and directories are a special kind of data storage resources. They are used for storing and providing metadata and other information about the resource of grid system. Information resources allow to store a huge structured amount of information about the current state of the grid system and to implement resources search tasks;
- **Network resource** is a link between the distributed resources of the grid system. The main characteristic of network resource is a data transfer rate [18].

The main areas of grid systems application include a scientific research, aerospace and automotive industry, architecture, electronics, energy, financial and banking infrastructure, medicine and biotechnology, manufacturing, media entertainment [19].

GRID – currently popular technology on the basis of GRID approaches have been created many applications, most of them have a strong mathematical direction (for example, a solution of large computational problems by dividing them into small pieces). A special case of such systems is Desktop-Grid – grid systems, in which the role of the compute nodes are the usual personal computers.

One of the examples can be a freely available Condor system, which is oriented to work with non-transferable (shared with the owner of the computer)

resources. The Condor system has the mechanisms of queuing work, policy planning, prioritization, taking into account the use and management of resources, support breakpoints, restart and migration tasks. The Condor developed very powerful resource description language that allows to formally describe the requirements for the types and amounts of resources on part of the job, as well as restrictions on the access to resources by the owners [20].

Also, there metacomputing system X-Com is designed for the rapid deployment and implementation of distributed calculations. The system is a tool to adapt and support the implementation of programs in distributed heterogeneous environments. At the heart of X-Com architecture is a client-server scheme. Accordingly, its application task must be logically divided into two parts: a server and a client. The server part is responsible for the decomposition of a problem into a plurality of independent portions and combining the results [21].

In general terms, the organization of such infrastructures can be identified by the four approaches: the first is based on the creation of projects that connect the executive computers; The second approach is the use of P2P-technology (Peer-to-Peer) and the union executive peer computers in the network; the third approach is characterized by systems with centralized management; the fourth approach is represented by private enterprise systems development.

The BOINC platform is designed to support projects but not available to run applications, as for each new application you need to create their own software and hardware infrastructure [22].

OurGrid - open and free to the association, the joint network, which is labs donate their inactive computing resources in exchange for the access to the resources of other inactive labs when necessary. It uses a peer technology that allows each facility to interact with the system [23].

XtremWeb is a software with an open code for creating grid systems of personal computers. Currently the system is being further developed under the name XWHEP. XWHEP architecture consists of three main types of components: servers, workers and customers. The Server provides the central services of the system, the

workers provide data resources within the system, which are installed by the owners on their computers. The Clients interact with the system and use its aggregated resources established by resource users on their computers. The system implements mechanisms for the access rights as a user-level, and user groups [24].

The weak points of systems that implement the approach with centralized management (for example, X-Com), a plan that does not allow the job to advance the specified period by the user. The need for pre-treatment job in X-Com also narrows the field of application of this system. Furthermore, in these developments, as well as systems CCOF and OurGrid, representing P2P-approach does not use the grid protocols which become standards. This leads to the fact that such developments are not interoperable with other grid systems. The same can be said about almost all corporate decisions.

Приложение Б

CD-ДИСК